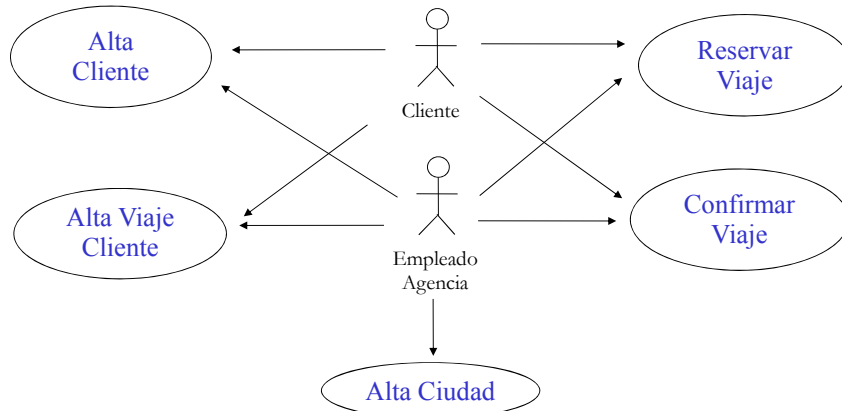


Modelo de Casos de Uso (Diagrama de Casos de Uso)



Modelo de Casos de Uso (Descripción de los casos de uso)

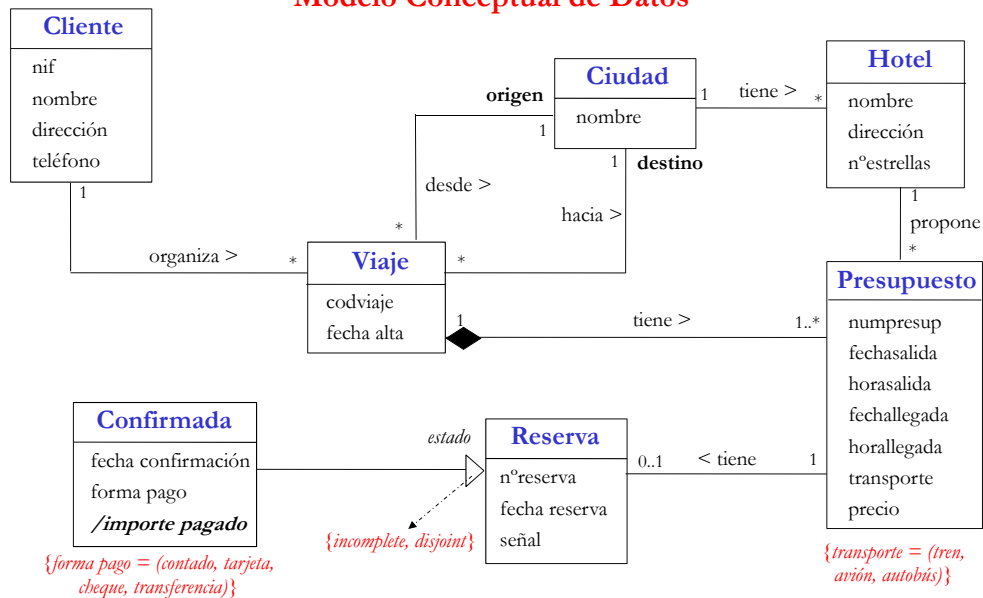
Caso de uso: Alta Viaje (resumen del escenario principal)

- El Empleado de la Agencia introduce los datos generales del viaje (*codviaje, fecha alta, ciudad origen y ciudad destino*) y el Sistema lo registra.
- El Empleado de la Agencia introduce los datos de los Presupuestos del Viaje y el Sistema los registra.

**Nota importante:**

El objetivo del caso de uso **Alta Viaje** es registrar en el sistema los datos generales del viaje y todos los presupuestos del mismo que existen en ese momento. Si se quiere que el sistema también tenga la funcionalidad de añadir un nuevo presupuesto a un viaje que ya se ha dado de alta, habría que añadir el caso de uso **“Añadir presupuesto”**.

## Modelo Conceptual de Datos



## Modelo Conceptual de Datos

## Restricciones de integridad:

- Claves de las clases:** (Cliente, nif), (Ciudad, nombre), (Hotel, nombre), (Viaje, codviaje), (Reserva, n°reserva)
- El hotel que se propone en un presupuesto tiene que estar en la ciudad de destino del viaje
- La ciudad origen de un viaje no puede ser también la ciudad de destino

## Reglas de derivación:

*/importe pagado* = precio del presupuesto del viaje reservado – señal pagada al reservar el viaje

## Modelo de Comportamiento del Sistema (DSS)

## Caso de Uso: Alta Viaje (escenario principal)



## Operación: IntroducirDatosViaje(w\_nif, w\_ciudadO, w\_ciudadD)

**Responsabilidades:** Registrar los datos generales del viaje en el sistema (generación y registro del código del viaje, fecha, cliente, ciudad origen y ciudad destino)

**Referencias cruzadas:** Caso de Uso: Alta Viaje

**Precondiciones:**

- Existe una ciudad con *nombre* = *w\_ciudadO* (*Co*)
- Existe una ciudad con *nombre* = *w\_ciudadD* (*Cd*)
- Existe un cliente con *nif* = *w\_nif* (*Cl*)

**Postcondiciones:**

- Se creó una instancia *V* de Viaje (creación de objeto)
- Se asignó *fecha\_sistema* a *V.fecha alta* y el *código del viaje generado por el sistema* a *V.codviaje* (modificación de atributos)
- Se asoció el objeto *Viaje V* con el objeto *Cl* de *Cliente* (creación de enlace)
- Se asoció el objeto *Viaje V* con el objeto *Co* de *Ciudad* (creación de enlace)
- Se asoció el objeto *Viaje V* con el objeto *Cd* de *Ciudad* (creación de enlace)

## Solución Problema 2 Modelo de Análisis en UML

7

**Operación: IntroducirPresupuesto** (*w\_fecha salida, w\_hora salida, w\_fecha llegada, w\_horallegada, w\_transporte, w\_hotel, w\_precio*)

**Responsabilidades:** Registrar un presupuesto del viaje (se genera el número del presupuesto y se registran los datos del presupuesto del viaje).

**Referencias cruzadas:** Caso de Uso: Alta Viaje

**Precondiciones:**

- Existe un Viaje **V** en curso
- Existe un objeto Hotel **H** con nombre igual a *w\_hotel* en la Ciudad de destino **Cd** del Viaje en curso **V**.

**Postcondiciones:**

- Se creó una instancia **P** de Presupuesto del Viaje **V** (creación de objeto)
- Se asignaron los datos del presupuesto a los correspondientes atributos del objeto *Presupuesto P* (*P.numero = número del presupuesto generado por el sistema, P.fecha salida = w\_fecha salida, P.fecha llegada = w\_fecha llegada, P.horallegada = w\_horallegada, P.transporte = w\_transporte, P.precio = w\_precio*) (modificación de atributos)
- Se creó un enlace de la asociación “*proponer*” entre el objeto Presupuesto **P** y el objeto Viaje **V** (creación de enlace)

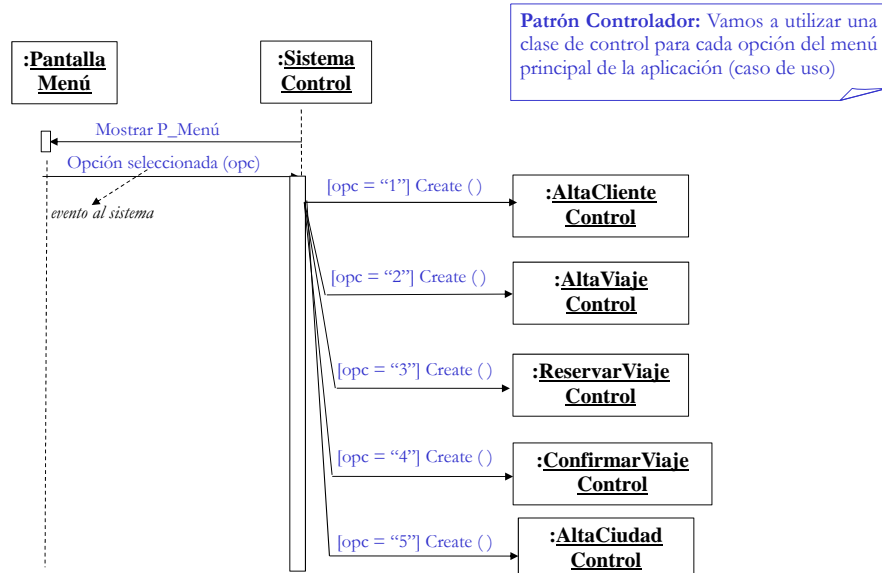
## Solución Problema 2 Modelo de Diseño en UML

8

- Al entrar en la aplicación se muestra el menú principal de la misma.
- En ese menú hay una opción para cada uno de los casos de uso del sistema.

Alta Cliente
Alta Viaje
Reservar Viaje
Confirmar Viaje
Alta Ciudad

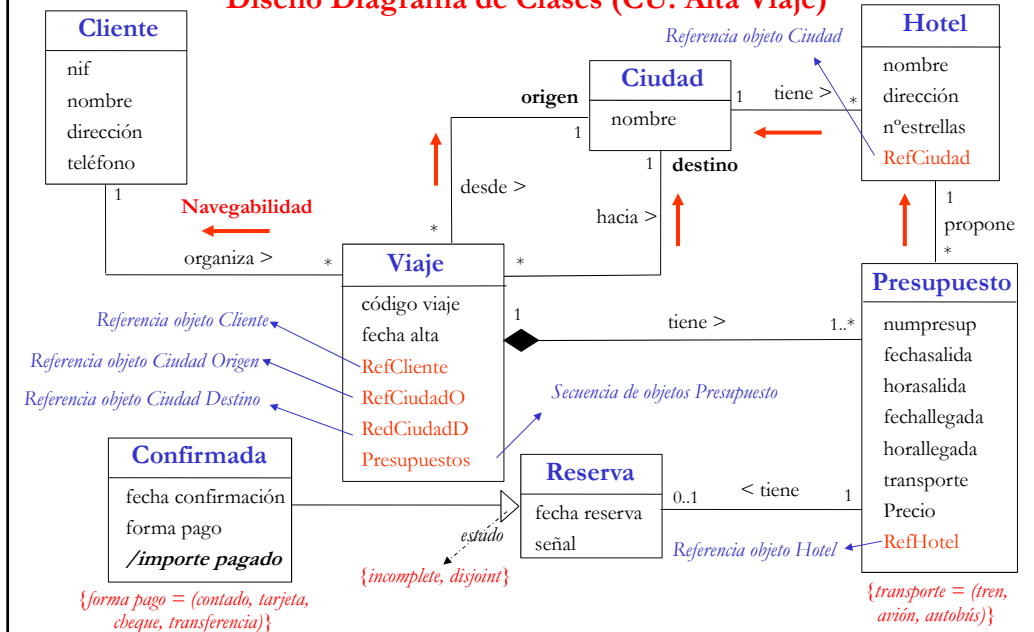
## Diagrama de Interacción Gestión Menú Principal



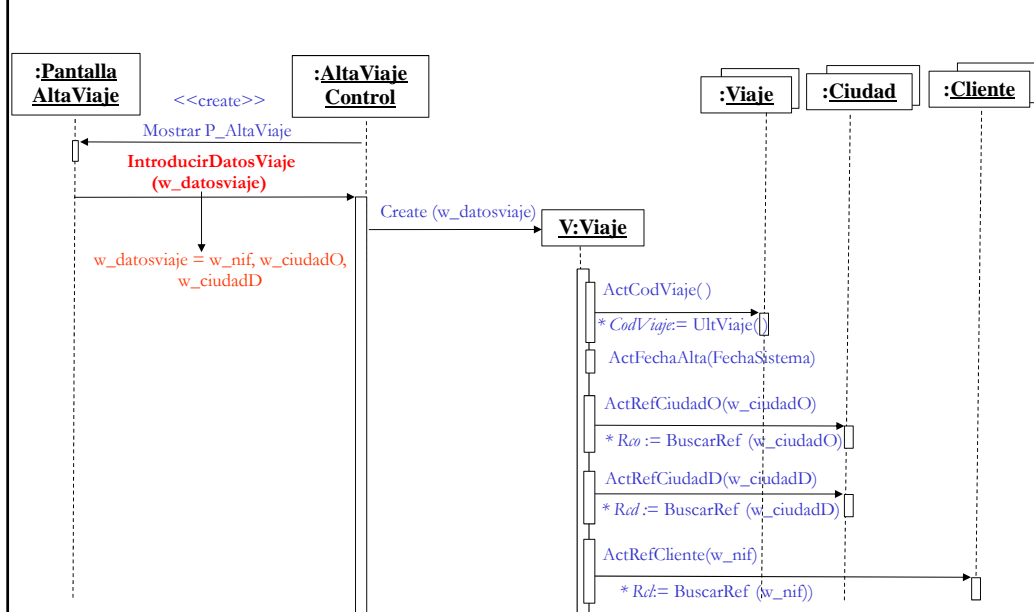
## Explicación Diagramas de Interacción para gestionar los menús

- Para ejecutar la aplicación, se crea un objeto de la clase *Sistema Control* (*new.SistemaControl*) y se ejecuta una operación de la clase *Sistema Control* que invoca a la operación *Mostrar P-menú* de la clase *Pantalla Menú*. La operación *Mostrar P-menú* se encarga de mostrar la pantalla del menú de la aplicación (en realidad esta operación invocará a otras operaciones de las clases de la parte de interfaz de usuario para mostrar la pantalla con el menú de la aplicación)
- Una vez mostrada la pantalla del menú de opciones, se está a la espera de que se produzca algún evento. En este caso el evento que se puede producir es que el usuario seleccione una determinada opción del menú (*Opción seleccionada (opc)*)
- En función de la opción del submenú que el usuario seleccione, se crea un objeto de la clase de control que controla la acción asociada a la opción (caso de uso) seleccionada. **Ejemplo:** si se selecciona la opción *Alta Cliente*, se crea un objeto de la clase de control *AltaClienteControl* (*new.AltaClienteControl*) y se ejecuta una operación definida en esta clase que le manda un mensaje a la Interfaz de Usuario (simplificada en la clase *Pantalla AltaCliente*) para que muestre la pantalla de Alta de Clientes

**Diseño Diagrama de Clases (CU: Alta Viaje)**



**Diagrama de Interacción Escenario Principal CU Alta Viaje**

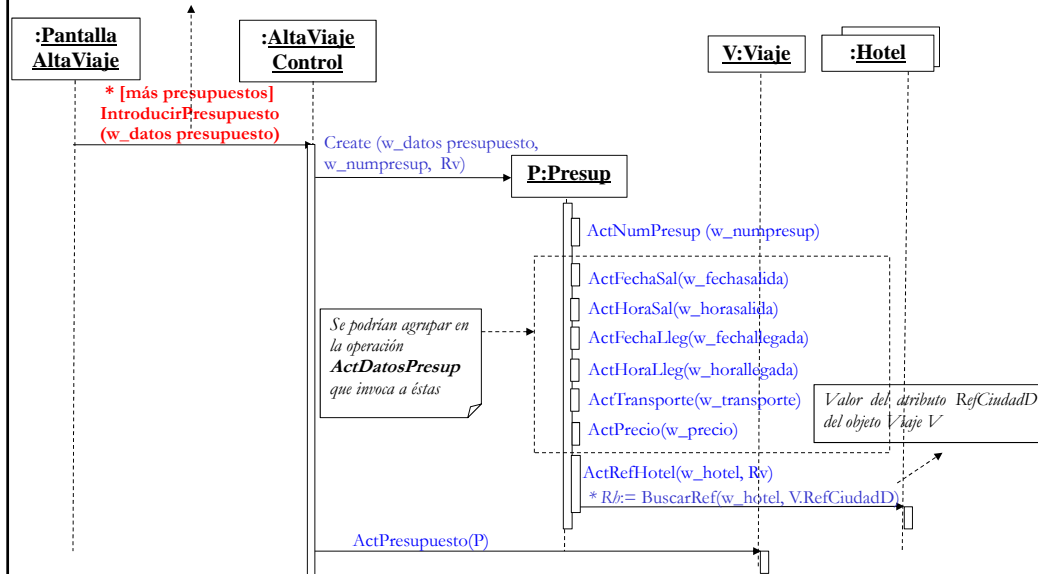


## Solución Problema 2 Modelo de Diseño en UML

13

### Diagrama de Interacción Escenario Principal CU Alta Viaje

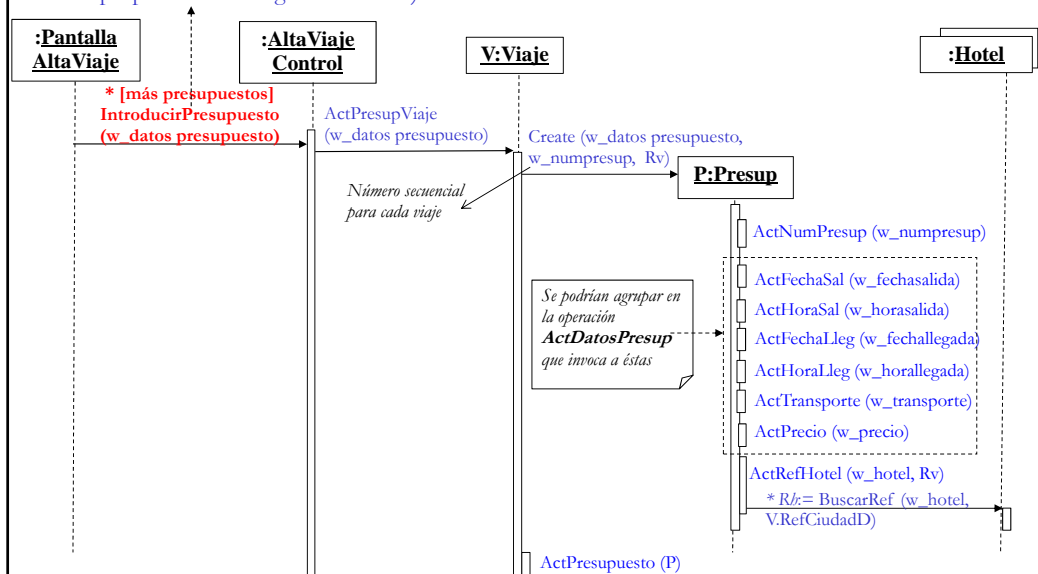
w\_datos presupuesto= w\_fechasalida, w\_horasalida, w\_fechallegada,  
w\_horallegada, w\_transporte, w\_hotel, w\_precio



## Solución Problema 2 Modelo de Diseño en UML

14

**Nota:** este diagrama de secuencia es otra forma alternativa de modelar la interacción asociada a la operación **IntroducirPresupuesto (w\_datos\_presupuesto)** que **mejora el acoplamiento y cohesión** respecto la solución propuesta en el diagrama anterior)



### Diagrama de Clases de Diseño (Clases)

En la solución que se propone de este problema, no se comprueban las precondiciones de los contratos de las operaciones antes de registrar los datos en el sistema y los datos se obtienen en el momento que se necesitan.

- De los diagramas de interacción que hemos hecho anteriormente, se obtienen nuevas clases, nuevas asociaciones y operaciones a añadir al diagrama de clases conceptual que hemos hecho en el análisis del sistema.
- **Clases nuevas:** clases de control y clases de la interfaz de usuario (simplificadas en clases Pantalla-X):
  - *Pantalla Menú*
  - *Sistema Control*
  - *AltaCliente Control*
  - *AltaViaje Control*
  - *ReservarViaje Control*
  - *ConfirmarViaje Control*
  - *AltaCiudad Control*
  - *Pantalla AltaViaje*

### Diagrama de Clases de Diseño (Asociaciones)

- **Colaboraciones entre objetos:**
  - *Pantalla Menú* y *Sistema Control*
  - *Sistema Control* y *AltaCliente Control*
  - *Sistema Control* y *AltaViaje Control*
  - *Sistema Control* y *ReservarViaje Control*
  - *Sistema Control* y *ConfirmarViaje Control*
  - *Sistema Control* y *AltaCiudad Control*
  - *Pantalla AltaViaje* y *AltaViaje Control*
  - *AltaViaje Control* y *Viaje*
  - *Viaje* y *Ciudad*
  - *Viaje* y *Cliente*
  - *Presupuesto* y *Hotel*



### Diagrama de Clases de Diseño (Operaciones)

- **Clase Sistema Control**
  - Operación que se ejecuta al hacer *new.SistemaControl*. Esta operación invoca a la operación *Mostrar P-menú* de la clase *Pantalla Menú*.
  - Operación *Opción seleccionada (opc)*. Invoca a la operación *Create ( )* de la clase de control asociada a la opción seleccionada.
- **Clase Pantalla Menú**
  - Operación *Mostrar P-menú*. Esta operación muestra la pantalla del menú principal de la aplicación.
- **Clase Pantalla Alta Viaje**
  - Operación *Mostrar P\_AltaViaje*. Muestra la pantalla para dar de alta viajes.

- **Clase Alta Viaje Control**
  - Operación que se ejecuta al hacer *new.AltViajeControl*. Invoca a la operación *Mostrar P\_AltaViaje* de la clase *Pantalla AltaViaje*.
  - Operación *IntroducirDatosViaje (w\_nif, w\_ciudadO, w\_ciudadD)*:
    - Invoca a la operación *Create (w\_datosviaje)* de *Viaje*.
  - Operación *IntroducirPresupuesto (w\_datospresupuesto)*:
    - Invoca a la operación *ActPresupViaje (w\_datospresupuesto)* del objeto *Viaje V*.

**Nota:** *w\_datospresupuesto* = *w\_fechasalida*, *w\_horasalida*, *w\_fechallegada*, *w\_horallegada*, *w\_transporte*, *w\_hotel*, *w\_precio*

- **Clase Viaje**

- Operación *Create (w\_datosviaje)* ( $w\_datosviaje = w\_nif, w\_ciudadO, w\_ciudadD$ )
  - Crea un objeto *V* de la clase Viaje
  - Invoca a la operación *ActCodViaje ()* del objeto Viaje *V*.
  - Invoca a la operación *ActFechaAlta (FechaSistema)* del objeto Viaje *V*.
  - Invoca a la operación *ActRefCiudadO (w\_ciudadO)* del objeto Viaje *V*.
  - Invoca a la operación *ActRefCiudadD (w\_ciudadD)* del objeto Viaje *V*.
  - Invoca a la operación *ActRefCliente (w\_nif)* del objeto Viaje *V*.
- Operación *ActPresupViaje (w\_datospresupuesto)*
  - Invoca a la operación *Create (w\_datospresupuesto, w\_numpresup, Rv )* de Presupuesto.
  - Invoca a la operación *ActPresupuesto (P)* del objeto Viaje *V*.

- **Clase Viaje (Continuación)**

- Operación *ActCodViaje ()*:
  - Invoca a la operación *\* CodViaje := UltViaje ()* de Viaje
  - $V.CodViaje := CodViaje + 1$
- Operación *ActFechaAlta(FechaSistema)*: asigna FechaSistema a V.fecha alta.
- Operación *ActRefCiudadO (w\_ciudadO)*:
  - Invoca a la operación *Rco := BuscarRef (w\_ciudadO)* de Ciudad
  - $V.RefCiudadO := Rco$
- Operación *ActRefCiudadD (w\_ciudadD)*:
  - Invoca a la operación *Rcd:= BuscarRef (w\_ciudadD)* de Ciudad
  - $V.RefCiudadD := Rcd$
- Operación *ActRefCliente (w\_nif)*:
  - Invoca a la operación *Rcl:= BuscarRef (w\_nif)* de Cliente
- Operación *\* CodViaje:=UltViaje ()*: Busca el código del último objeto Viaje creado.
- Operación *ActPresupuesto (P)*: añade el objeto presupuesto P a V.Presupuestos

- **Clase Ciudad**

- Operación *\* Rc:=BuscarRef (w\_cidad)*:
  - Busca la referencia del objeto Ciudad con *nombre = w\_cidad*. Esta operación se invoca para buscar la referencia de la ciudad origen y la referencia de la ciudad destino.

- **Clase Cliente**

- Operación *Rcl := BuscarRef (w\_nif)*:
  - Busca la referencia del objeto Cliente con *nif = w\_nif*.

- **Clase Hotel**

- Operación *\* Rh:= BuscarRef (w\_hotel, V.RefCiudadD)*
  - Busca la referencia del objeto hotel con *nombre = w\_hotel* y con *RefCiudad = V.RefCiudadD*.

- **Clase Presupuesto**

- Operación *Create (w\_datospresupuesto, w\_numpresup, Rv)*
  - Crea un objeto *P* de la clase Presupuesto.
  - Invoca a la operación *ActNumPresup (w\_numpresup)* del presupuesto *P*.
  - Invoca a las operaciones de asignación de valores a los atributos del objeto Presupuesto *P*:  
*ActFechaSal(w\_fechaSalida)*, *ActHoraSal (w\_borasalida)*, *ActFechaLeg(w\_fechaLlegada)*, *ActHoralle (w\_borallegada)*, *ActTransporte (w\_transporte)*, *ActPrecio (w\_precio)*  
 (También se podría añadir la operación *ActDatosPresup* que invoca a todas estas)
  - Invoca a la operación *ActRefHotel (w\_hotel, Rv)* de presupuesto *P*.

- **Clase Presupuesto (Continuación)**

- Operación *ActRefHotel* (*w\_hotel*, *Rv*)
  - Invoca a la operación \* *Rh* := *BuscarRef* (*w\_hotel*, *V.RefCiudadID*).
  - *P.RefHotel* = *Rh*.
- Operaciones de asignación de valores al resto de los atributos del objeto Presupuesto *P*:
  - Operación *ActFechaSal* (*w\_fechaSalida*): *P.fechaSalida* = *w\_fechaSalida*
  - Operación *ActHoraSal* (*w\_horaSalida*): *P.horaSalida* = *w\_horaSalida*.
  - Operación *ActHoraLeg* (*w\_horaLlegada*): *P.horaLlegada* = *w\_horaLlegada*.
  - Operación *ActTransporte* (*w\_transporte*): *P.transporte* = *w\_transporte*.
  - Operación *ActPrecio* (*w\_precio*): *P.precio* = *w\_precio*.