



INGENIERÍA DEL SOFTWARE I

Práctica 6

Modelado de Diseño

Univ. Cantabria – Fac. de Ciencias
María Sierra y Patricia López



Clases de Diseño

- Son clases completamente definidas, en las que su estructuras de datos internas quedan perfectamente definidas, así como los diagramas de estados y actividades que especifican las operaciones y métodos de su interfaces.

Orden
FechaSolicitud
FechaEntrega
NumeroOrden
Calcular el precio total()
Registrar los impuestos de gestión de la orden()
Registrar el número de orden()

Orden
-FechaSolicitud : Date
-fechaEntrega : Date
-numeroOrden : int
-impuestos : float
-precioTotal : float
#calcularImpuesto(gstis : parameter) : float
#calcularTotal() : float
+getImpuesto() : float
+setNumeroOrden(nOrden : int)
+getNumeroOrden() : int

- **Visibilidad**
 - **Atributos y Tipos**
 - **Métodos:**
 - Parámetros
 - Valores retorno
 - Flujo de ejecución (diagramas de actividad) o incluso pseudocódigo (campo Semantics de la especificación de la operación)
- **Identificación de clases activas:** aquellas que necesitan su propio flujo de control dentro del sistema (considerando los requisitos de concurrencia del mismo).



Introducción al Diseño

- **Modelamos el sistema** y definimos su forma (incluida la arquitectura) para que soporte todos los requisitos (incluidos requisitos no funcionales y otras restricciones).
- Se optimiza el modelo de análisis
 - Se define el “como” va a ser implementado el modelo de análisis
 - Se tienen en cuenta aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, SO, distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de almacenamiento de datos, etc..
- Punto de partida para la implementación:
 - Los programadores deben poder implementar el diseño sin que surjan dudas esenciales.
- Un diseño es una solución particular de las muchas posibles.



Diseño de la Arquitectura

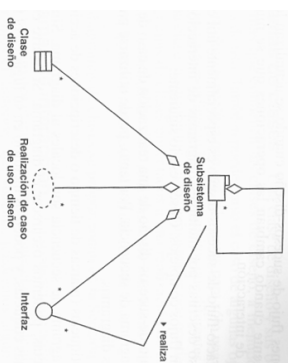
- **Identificación de subsistemas y sus interfaces**
 - Permiten **organizar** el modelo de diseño en piezas manejables.
 - Se identifican a partir de los **paquetes de análisis**, pero puede ser necesario un refinamiento para tratar temas relativos al diseño, implementación y distribución del sistema:
 - Una parte de un anterior paquete de análisis, compartida y utilizada por varios subsistemas
 - Algunas partes de un anterior paquete de análisis se realizan por **productos SW reutilizados** (asignables a capas intermedias o subsistemas de SW del sistema)
 - Los anteriores paquetes del análisis **no permiten la incorporación de un sistema heredado**
 - Puede que la descomposición de subsistemas tenga que tratar los aspectos de distribución, descomponiéndolos en subsistemas más pequeños de forma que cada uno pueda asignarse a un nodo determinado.
- En algunos casos, **no todos los subsistemas se implementan** ya que algunos representan **productos reutilizados** y otros son **recursos ya existentes**. Su inclusión en el modelo de diseño permite analizar y evaluar las alternativas de reutilización.



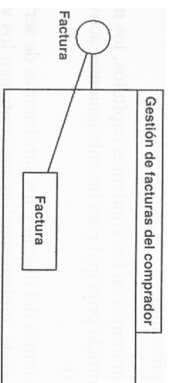
Subsistemas de Diseño

Constan de:

- clases del diseño,
- realizaciones de casos de uso, interfaces
- otros subsistemas (recursivamente)



- **Puede** proporcionar **interfaces** que representen la funcionalidad que exporta en términos de operaciones
- Las operaciones pueden ser implementadas por alguna de las clases internas (o por colaboraciones entre ellas).

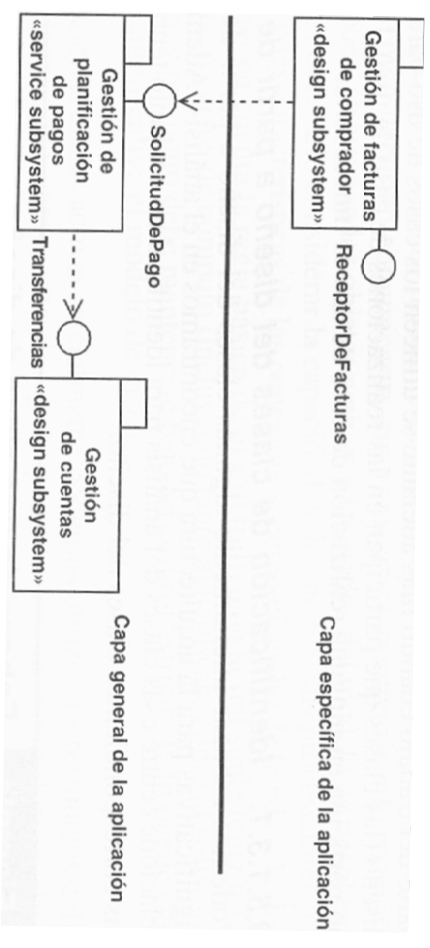


María Sierra, Patricia López - ISI

P6.5



Subsistemas de diseño



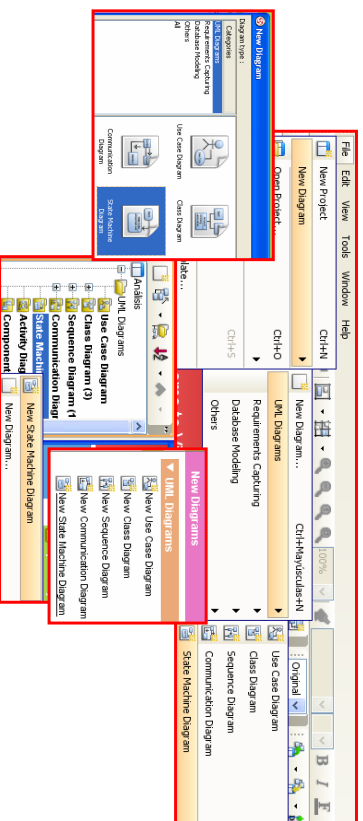
María Sierra, Patricia López - ISI

P6.6



Diagramas de Estado con VP

Crear Diagrama



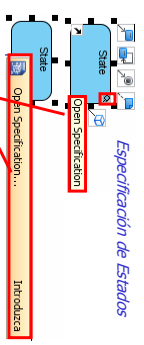
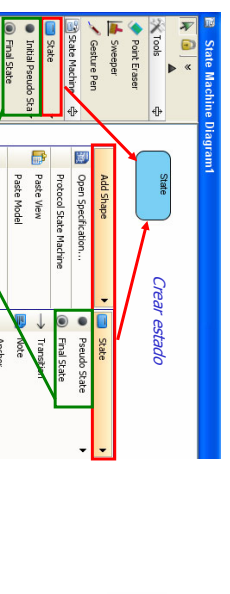
María Sierra, Patricia López - ISI

P6.7



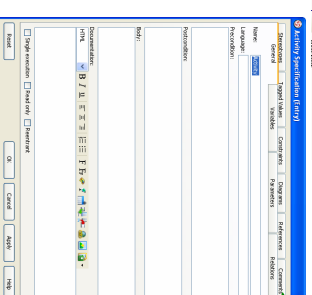
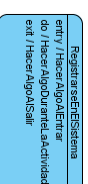
Diagramas de Estado con VP: Elementos

Estado



Acciones de Estado de entrada/salida

Acciones que se ejecutan independientemente de los eventos que lleguen a los estados



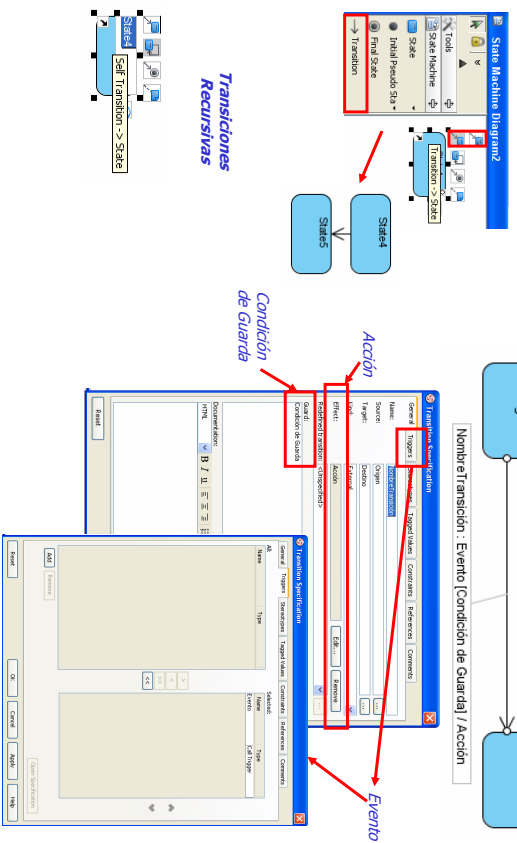
María Sierra, Patricia López - ISI

P6.8



Diagramas de Estado con VP: Elementos

Transiciones

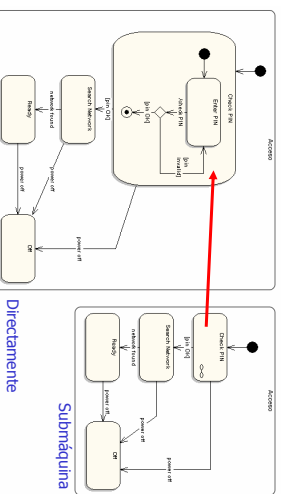


María Sierra, Patricia López - ISI



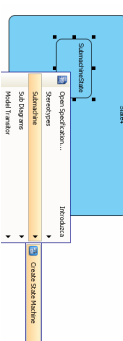
Diagramas de Estado con VP: Elementos

Estados Compuestos: Subestados

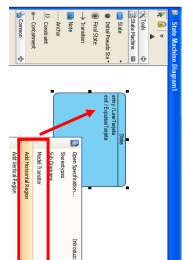


- Un diagrama de máquina de estado puede incluir diagramas de submáquinas
- Mediante estados de Submáquina: los detalles se muestran en un diagrama separado
- Directamente

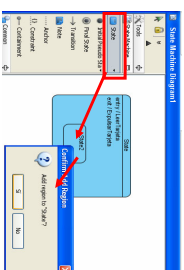
Mediante Submáquina creando diagrama



Añadimos región y luego elementos



Añadimos un estado y una región



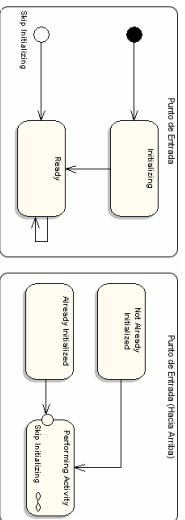
María Sierra, Patricia López - ISI



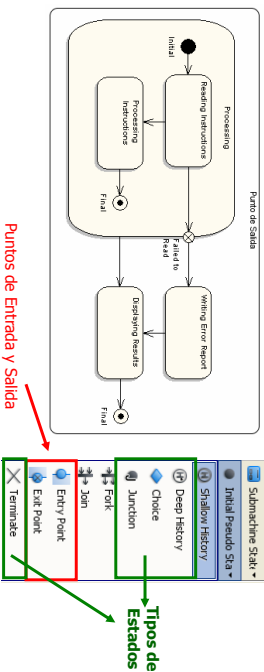
Diagramas de Estado con VP: Elementos

Puntos de Entrada

Algunas veces no desearemos ingresar a una submáquina en un Estado Inicial normal. Por ejemplo, en la siguiente submáquina sería normal comenzar en el estado inicial, pero si por alguna razón no fuera necesario realizar la inicialización, sería posible comenzar en el estado ready realizando una transición al punto de entrada nombrado



Puntos de Entrada y Salida



Puntos de Salida

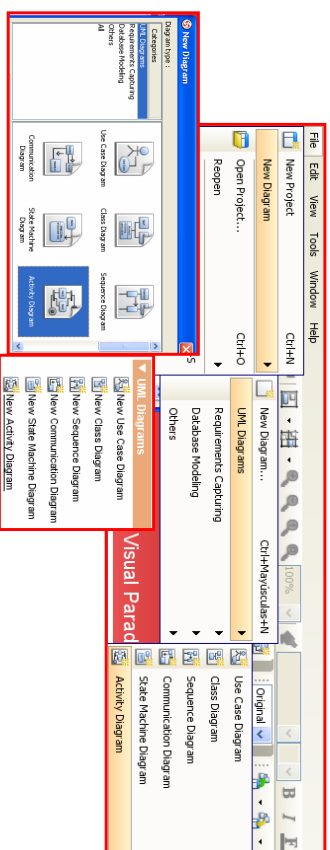
Es posible nombrar Puntos de Salida distintos del normal. El siguiente diagrama provee un ejemplo donde el estado ejecutado después del estado de procesos principal depende de que ruta se use para realizar la transición del estado

María Sierra, Patricia López - ISI



Diagramas de Actividad con VP

Crear Diagrama



María Sierra, Patricia López - ISI

