

PROGRAMACION I

Nivel 1

Problemas, Soluciones y Programas

Universidad Pedagógica y Tecnológica de Colombia

Escuela de Ingeniería de Sistemas – UniAndes

Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Análisis del
problema

2

Diseño de la
solución

3

Construcción de
la solución =
Implementación =
Programación

Solución



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes



- **Entender** el problema que tiene el cliente
- **Especificar** TODA la información que suministre el cliente



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Problema

Análisis del problema

2

Diseño de la solución

3

Construcción de la solución =
Implementación =
Programación

Solución



- **Detallar** las características que tendrá la solución

- Usando algún **lenguaje** (planos, dibujos, ecuaciones, **diagramas**, texto, ...)



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Problema

Análisis del problema

2

Diseño de la solución

3

Construcción de la solución =
Implementación =
Programación

Solución

- **Implementar** el programa a partir del diseño

- **Probar** su correcto funcionamiento



Caso de Estudio: El Empleado



El Empleado

- Se quiere una aplicación que permita manejar la información de un empleado.
- El empleado tiene:
 - Nombre
 - Apellido
 - Sexo
 - Fecha de nacimiento
 - Imagen asociada
 - Fecha de ingreso a la misma
 - Salario básico asignado
- La aplicación debe permitir:
 - Modificar el salario del empleado
 - Realizar algunos cálculos con la información disponible
 - Edad actual
 - Antigüedad en la empresa
 - Prestaciones a las que tiene derecho. Para el cálculo de las prestaciones se utiliza la fórmula $p = (a * s)/12$ (p: prestaciones, a: antigüedad, s: salario).



El Empleado

Sistema de Empleados

Datos Personales


Nombre:

Apellido:

Sexo:

Fecha de Nacimiento:

Fecha de Ingreso:



Salario

Salario:

Cálculos

Puntos de Extensión



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Problema

Análisis del problema

2

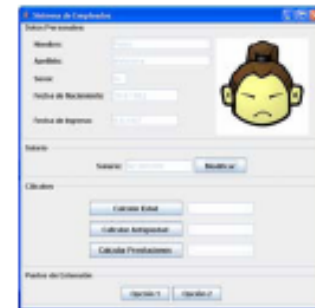
Diseño de la solución

3

Construcción de la solución =
Implementación =
Programación

Solución

Se quiere una aplicación que permita manejar la información de un empleado.

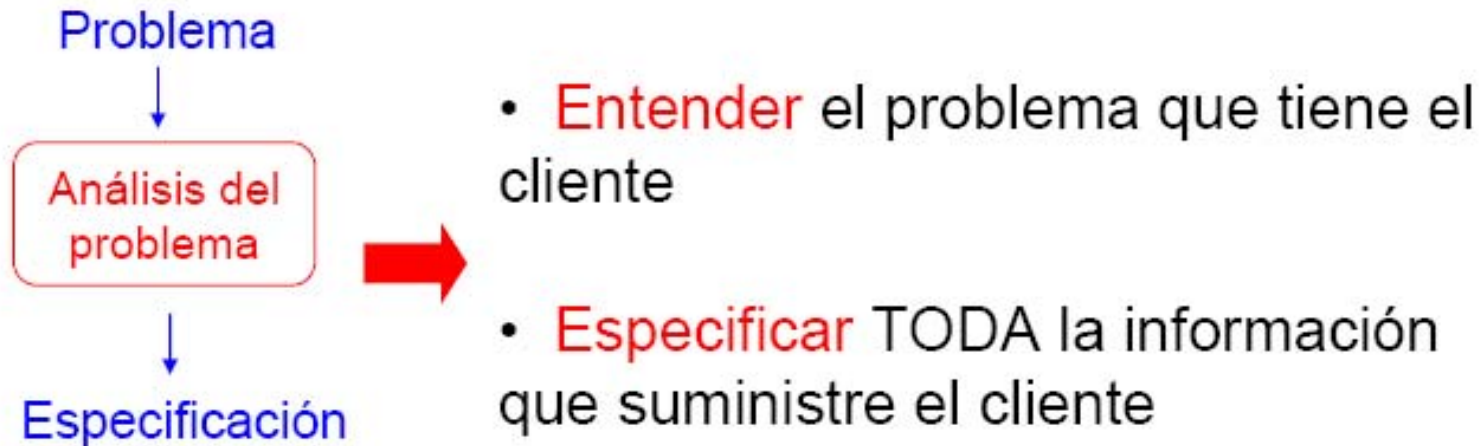


1

Análisis del Problema



Análisis del Problema



Qué quiere decir ESPECIFICACION ...



Especificación



Requerimientos
funcionales (RF)

Mundo del
Problema

Requerimientos
No Funcionales



Especificación



Requerimientos
funcionales (RF)



- Las **necesidades** del cliente
- **Operaciones** o **servicios** que el programa debe proveer al usuario



El Empleado

- Se quiere una aplicación que permita manejar la información de un empleado.
- El empleado tiene:
 - Nombre
 - Apellido
 - Sexo
 - Fecha de nacimiento
 - Imagen asociada
 - Fecha de ingreso a la misma
 - Salario básico asignado
- La aplicación debe permitir:
 - Modificar el salario del empleado
 - Realizar algunos cálculos con la información disponible
 - Edad actual
 - Antigüedad en la empresa
 - Prestaciones a las que tiene derecho. Para el cálculo de las prestaciones se utiliza la fórmula $p = (a * s)/12$ (p: prestaciones, a: antigüedad, s: salario).

Requerimientos
funcionales (RF)



Requerimientos Funcionales

- **R1:** Modificar el salario del empleado.
- **R2:** Calcular la edad actual.
- **R3:** Calcular la antigüedad en la empresa.
- **R4:** Calcular las prestaciones.



El Empleado

Sistema de Empleados

Datos Personales


Nombre:

Apellido:

Sexo:

Fecha de Nacimiento:

Fecha de Ingreso:



Salario

Salario:

Cálculos

Puntos de Extensión

R1 points to the 'Modificar' button in the Salario section.

R2 points to the 'Calcular Edad' button in the Cálculos section.

R3 points to the 'Calcular Antigüedad' button in the Cálculos section.

R4 points to the 'Calcular Prestaciones' button in the Cálculos section.



Especificación



Mundo del
Problema



- **Contexto** en el que ocurre el problema
- Elementos (**datos, información**) que intervienen en el problema



El Empleado

- Se quiere una aplicación que permita manejar la información de un empleado.
- El empleado tiene:
 - Nombre
 - Apellido
 - Sexo
 - Fecha de nacimiento
 - Imagen asociada
 - Fecha de ingreso a la misma
 - Salario básico asignado
- La aplicación debe permitir:
 - Modificar el salario del empleado
 - Realizar algunos cálculos con la información disponible
 - Edad actual
 - Antigüedad en la empresa
 - Prestaciones a las que tiene derecho. Para el cálculo de las prestaciones se utiliza la fórmula $p = (a * s)/12$ (p: prestaciones, a: antigüedad, s: salario).



Mundo del
problema



Especificación



Requerimientos
No Funcionales

- **Restricciones** o condiciones que impone el cliente al programa
- Ejemplos: Tiempo de entrega del programa, # de usuarios simultáneos, tiempo de ejecución del programa, ...



1

Análisis del Problema



Especificación de
Requerimientos Funcionales



Requerimientos Funcionales

- Se describen a través de 4 elementos:
 - **Identificador** y **nombre**
 - **Resumen** de la operación
 - **Entradas** que debe dar el **usuario** para que el programa pueda realizar la operación
 - **Resultado** de la operación
 - **Modificación** de un valor en el mundo del problema
 - **Cálculo** de un valor
 - **Mezcla** de los dos anteriores



Ejemplo RF

Nombre	R4 – Calcular las prestaciones del empleado
Resumen	Calcula las prestaciones del empleado
Entradas	
Resultados	



Ejemplo RF

Nombre	R1 – Actualizar el salario básico del empleado
Resumen	Permite la modificación del salario básico de un empleado
Entradas	
Nuevo salario	
Resultados	
Se modificó el salario básico del empleado	



Caso de Estudio: El Simulador Bancario



El Simulador bancario

- Se quiere una aplicación que haga la simulación en el tiempo de la **cuenta bancaria** de un **cliente**.
- Un **cliente** tiene:
 - Nombre
 - Número de cédula (identifica la cuenta)
- Una **cuenta** tiene:
 - Una cuenta de ahorro
 - Una cuenta corriente
 - Certificado de depósito a término (CDT)
- Se quiere que el programa permita a una persona simular el manejo de sus productos bancarios:
 - Hacer las operaciones necesarias sobre los productos que conforman la cuenta
 - Avanzar mes por mes en el tiempo, para que el cliente pueda ver el resultado de sus movimientos bancarios y el rendimiento de sus inversiones



El Simulador Bancario

Simulador Bancario

Datos Personales

Nombre: Cédula: Mes:

Saldo

Saldo Corriente:

Saldo Ahorros:

Saldo CDT: Total:

Cálculos

<input type="button" value="Abrir inversion CDT"/>	<input type="button" value="Consignar cuenta corriente"/>	<input type="button" value="Consignar cuenta ahorro"/>	<input type="button" value="Opcion1"/>
<input type="button" value="Cerrar inversion CDT"/>	<input type="button" value="Retirar cuenta corriente"/>	<input type="button" value="Retirar cuenta ahorro"/>	<input type="button" value="Opcion2"/>



1

Análisis del Problema



Mundo del Problema



Modelo del mundo del problema

- Esta actividad está basada en un proceso de “observación” del problema.
- El objetivo es identificar los elementos que allí aparecen y describirlos de la mejor manera.
- Cuatro actividades para llevar a cabo esta etapa:



Identificar las entidades

Modelar las características de las entidades

Buscar las relaciones entre las entidades

Documentar (reglas, restricciones,
etc.)

MODELO DEL MUNDO

Para expresar el modelo del mundo, usamos diagramas de clases del lenguaje de modelos UML (estándar).



Ejercicio

- Identifique los elementos que hacen parte del mundo del problema en el simulador bancario
 - Cuenta bancaria
 - Cuenta de ahorros
 - Cuenta corriente
 - CDT
 - Cliente
 - Cedula
 - Nombre
 - Mes (de simulación)
 - Saldo cuenta corriente
 - Saldo cuenta ahorro
 - Saldo CDT
 - Saldo total
 - Intereses CDT
 - Interés cuenta de ahorro
 - Inversión CDT (monto)



Modelo del mundo del problema

- Esta actividad está basada en un proceso de “observación” del problema.
- El objetivo es identificar los elementos que allí aparecen y describirlos de la mejor manera.
- Cuatro actividades para llevar a cabo esta etapa:



Identificar las entidades

Modelar las características de las entidades

Buscar las relaciones entre las entidades

Documentar (reglas, restricciones,
etc.)

MODELO DEL MUNDO

Para expresar el modelo del mundo, usamos diagramas de clases del lenguaje de modelos UML (estándar).



Identificar las Entidades

- Elementos relevantes del mundo que intervienen en el problema
 - Concretos (persona, vehículo)
 - Abstractos (cuenta bancaria)
- Se les da un nombre significativo
- Pista para ubicarlos: sustantivos del problema
- En POO las llamamos **CLASES**
- Convención: los nombres de las clases empiezan por mayúscula



Elementos del mundo del problema

- Cuenta bancaria
- Cuenta de ahorros
- Cuenta corriente
- CDT
- Cliente
- Cedula
- Nombre
- Mes (de simulación)
- Saldo cuenta corriente
- Saldo cuenta ahorro
- Saldo CDT
- Saldo total
- Intereses CDT
- Interés cuenta de ahorro
- Inversión CDT (monto)



Elementos del mundo del problema

- Cuenta bancaria
- Cuenta de ahorros
- Cuenta corriente
- CDT
- Cliente
- Cedula
- Nombre
- Mes (de simulación)
- Saldo cuenta corriente
- Saldo cuenta ahorro
- Saldo CDT
- Saldo total
- Intereses CDT
- Interés cuenta de ahorro
- Inversión CDT (monto)

NO TODOS ESTOS ELEMENTOS SON CLASES



Clases del simulador bancario

- Cuenta bancaria
- Cuenta de ahorros
- Cuenta corriente
- CDT
- Cliente
- Cedula
- Nombre
- Mes (de simulación)
- Saldo cuenta corriente
- Saldo cuenta ahorro
- Saldo CDT
- Saldo total
- Intereses CDT
- Interés cuenta de ahorro
- Inversión CDT (monto)

**TODOS LOS DEMAS ELEMENTOS DEL PROBLEMA
SON CARACTERISTICAS DE LAS CLASES**



Clases del simulador bancario

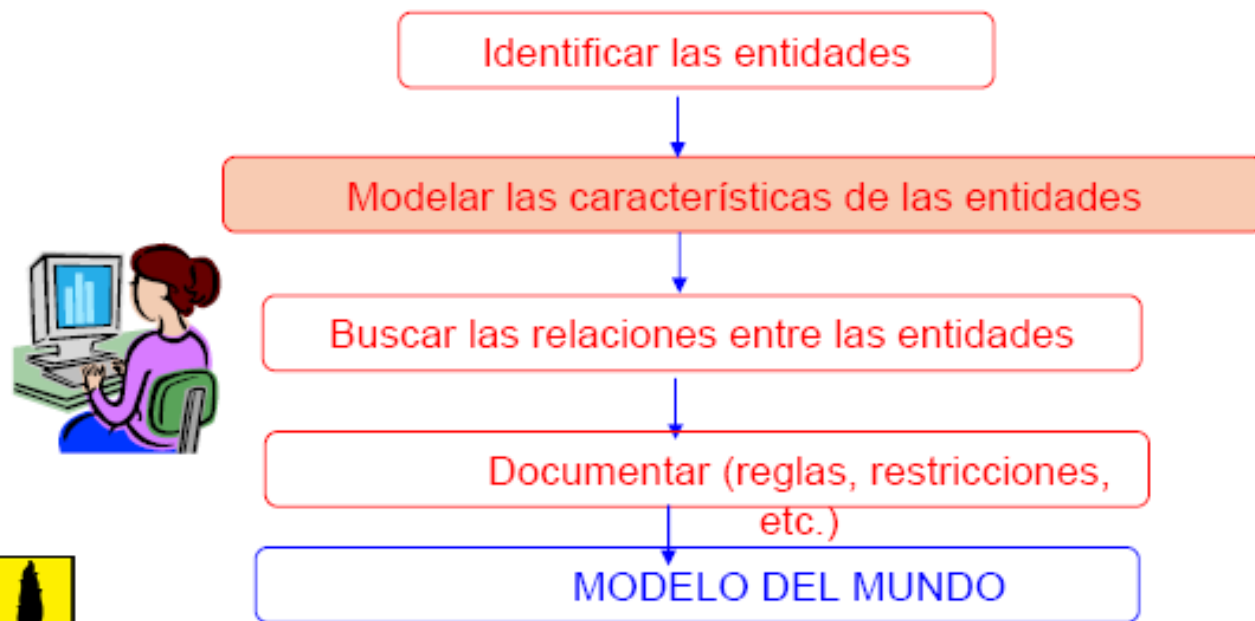
- CuentaBancaria
- CuentaCorriente
- CuentaAhorros
- CDT
- Mes

Hacen parte de la CuentaBancaria



Modelo del mundo del problema

- Esta actividad está basada en un proceso de “observación” del problema.
- El objetivo es identificar los elementos que allí aparecen y describirlos de la mejor manera.
- Cuatro actividades para llevar a cabo esta etapa:



Para expresar el modelo del mundo, usamos diagramas de clases del lenguaje de modelos UML (estándar).



Modelar las características

- A cada característica le debemos asociar:
 - Nombre significativo
 - Descripción del conjunto de valores que dicha característica puede tomar
- En POO las llamamos **ATRIBUTOS**
- Convención: los nombres de los atributos empiezan por minúscula, sin espacios en blanco



Ejemplo sobre el Empleado

Sistema de Empleados

Datos Personales


Nombre:

Apellido:

Sexo:

Fecha de Nacimiento:

Fecha de Ingreso:



Salario

Salario:

Cálculos

Puntos de Extensión



Ejemplo sobre el Empleado

Atributo	Valores posibles
nombre	Cadena de caracteres
apellido	Cadena de caracteres
sexo	Masculino o Femenino
salario	Valores enteros positivos

Para expresar el modelo del mundo, usamos diagramas de clases del lenguaje de modelos UML (estándar).

Características o atributos

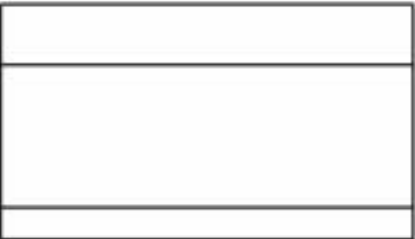


nombre de la entidad o clase

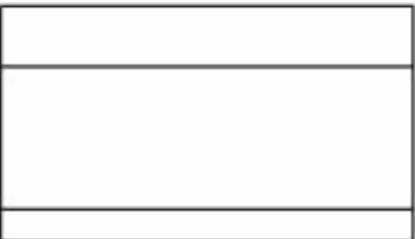


Ejercicio sobre el Simulador Bancario

Clase: CuentaBancaria

Atributo	Valores posibles	Diagrama UML
		

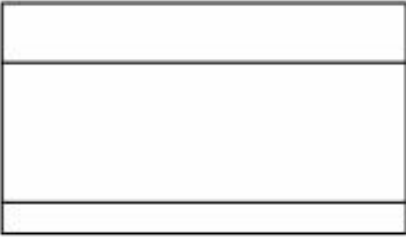
Clase: CuentaCorriente

Atributo	Valores posibles	Diagrama UML
		



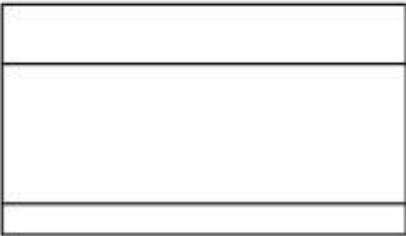
Ejercicio sobre el Simulador Bancario

✚ Clase: CDT

Atributo	Valores posibles	Diagrama UML
		



Clase: Mes

Atributo	Valores posibles	Diagrama UML
		



Modelo del mundo del problema

- Esta actividad está basada en un proceso de “observación” del problema.
- El objetivo es identificar los elementos que allí aparecen y describirlos de la mejor manera.
- Cuatro actividades para llevar a cabo esta etapa:



Identificar las entidades

Modelar las características de las entidades

Buscar las relaciones entre las entidades

Documentar (reglas, restricciones,
etc.)

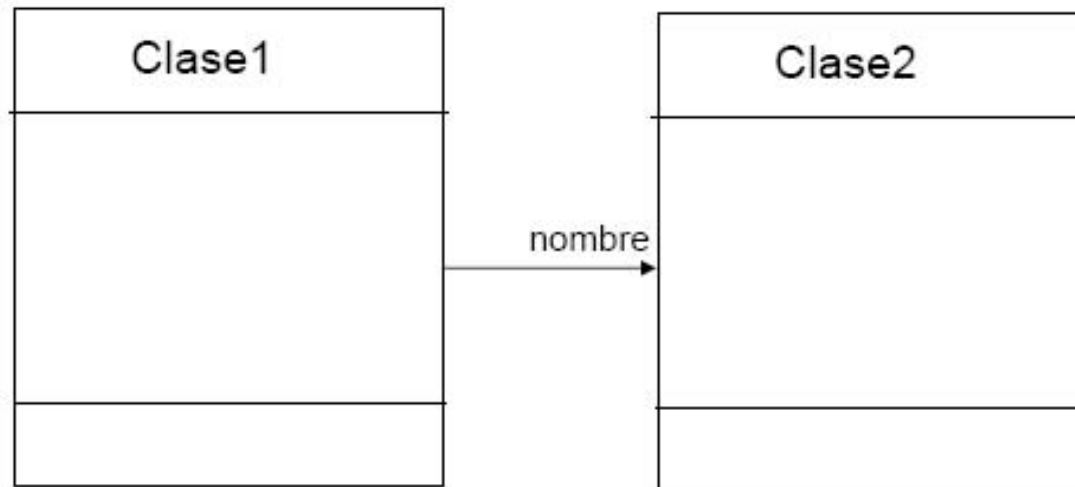
MODELO DEL MUNDO

Para expresar el modelo del mundo, usamos diagramas de clases del lenguaje de modelos UML (estándar).



Las relaciones entre las Entidades

- Identificar las relaciones que existen entre las distintas entidades del mundo (clases).
- Dar un nombre a cada relación
- En POO las llamamos **ASOCIACIONES**

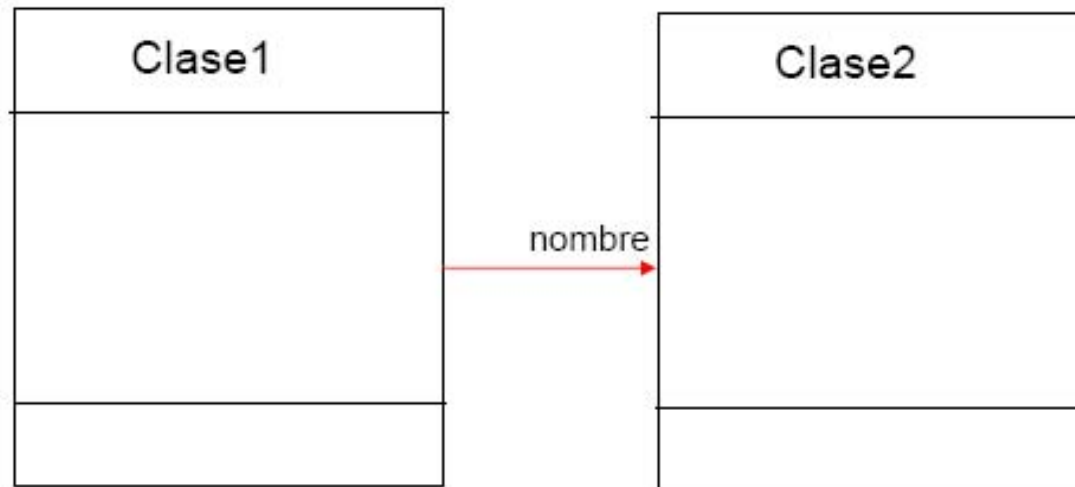


En
UML



Las relaciones entre las Entidades

- Identificar las relaciones que existen entre las distintas entidades del mundo (clases).
- Dar un nombre a cada relación
- En POO las llamamos **ASOCIACIONES**



En
UML



Ejemplo sobre el Empleado

Sistema de Empleados

Datos Personales


Nombre:

Apellido:

Sexo:

Fecha de Nacimiento:

Fecha de Ingreso:



Salario

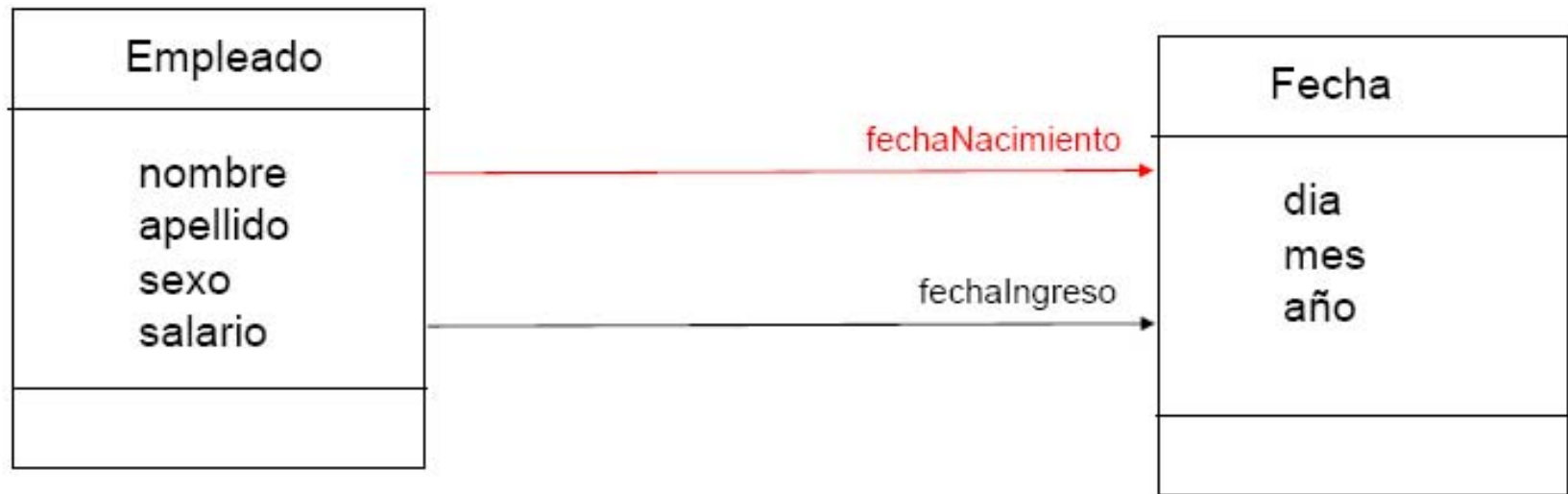
Salario:

Cálculos

Puntos de Extensión



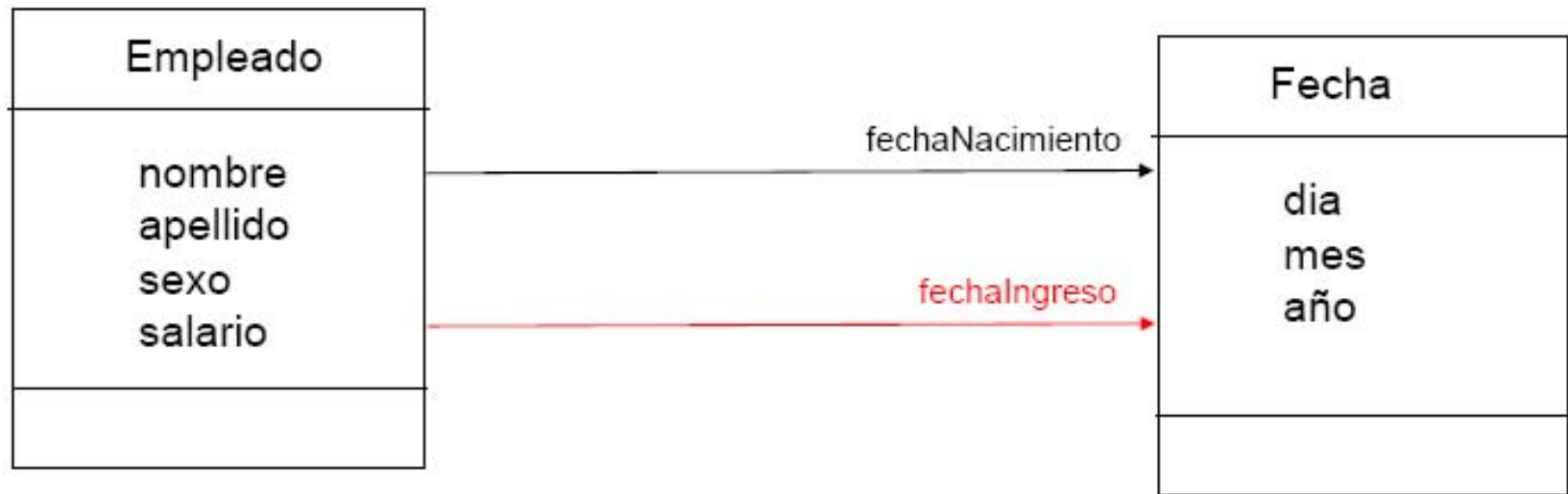
Ejemplo sobre el Empleado



- El empleado **tiene** una fecha de nacimiento
- Esta fecha es una entidad del mundo representada por la clase **Fecha**



Ejemplo sobre el Empleado



- El empleado **tiene** una fecha de ingreso
- La dirección de la flecha indica que la clase que **contiene** a la otra:
 - El empleado **tiene** una fecha, pero la fecha NO tiene un empleado



Diagrama de clases del modelo del mundo para el simulador bancario

CuentaBancaria

Mes

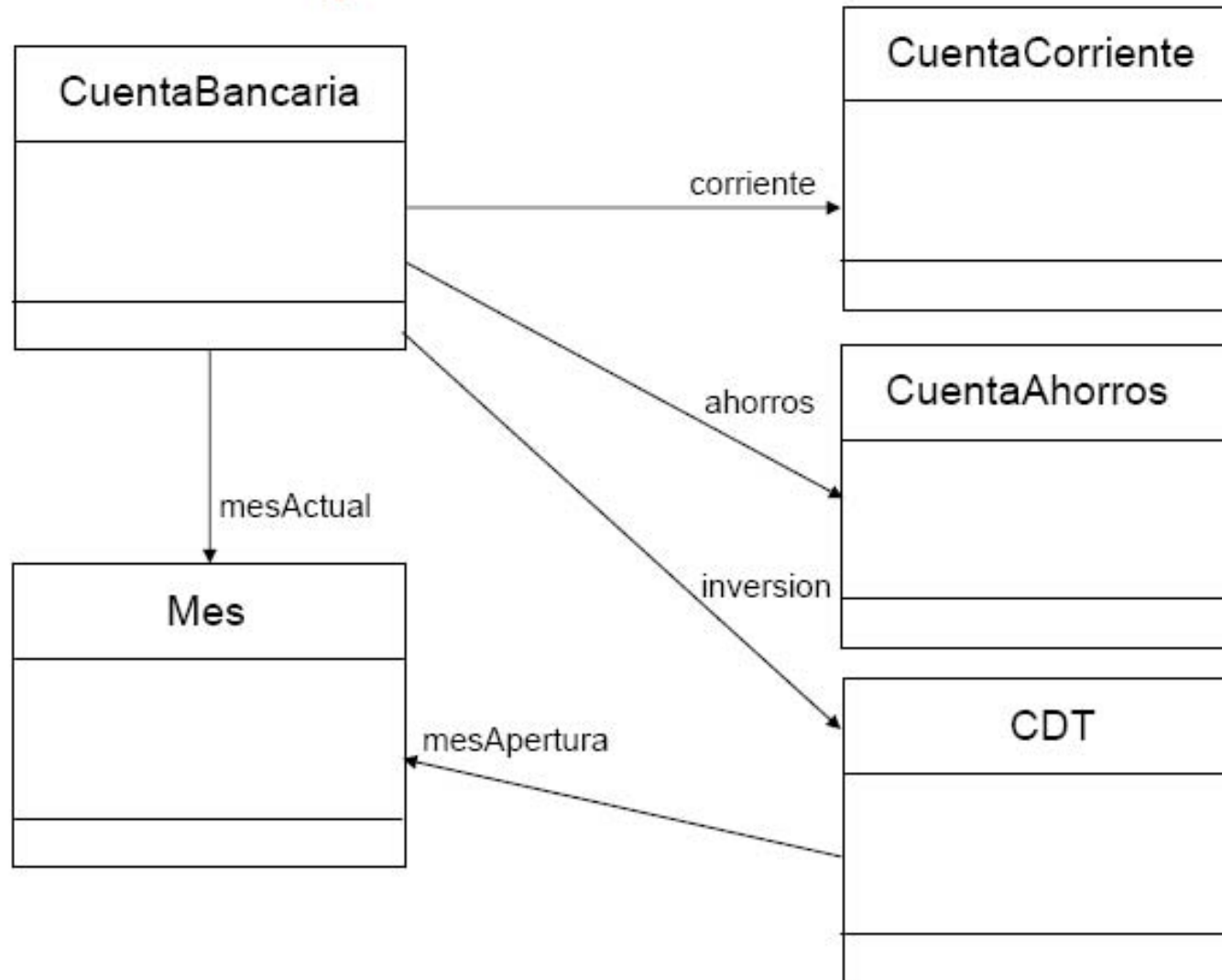
CuentaCorriente

CuentaAhorros

CDT



Diagrama de clases del modelo del mundo para el simulador bancario



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Problema

Análisis del problema

2

Diseño de la solución

3

Construcción de la solución =
Implementación =
Programación

Solución



- **Detallar** las características que tendrá la solución

- Usando algún **lenguaje** (planos, dibujos, ecuaciones, **diagramas**, texto,

...) **UML**



Solucionar un Problema = Construir un Programa



Programador

Herramientas y Lenguajes

1

Problema

Análisis del problema

2

Diseño de la solución

3

Construcción de la solución =
Implementación =
Programación

Solución

- **Implementar** el programa a partir del diseño

- **Probar** su correcto funcionamiento



3

Construcción de la Solución

