

## Reglas de Negocio: ¿Dónde implementarlas?



En el servidor: mediante PK, FK, unique, Check...  
disparadores, procedimientos, funciones, paquetes.

☺ Garantizan la consistencia de los datos almacenados.

Si sólo las implementamos en el servidor:

- ☹ Frustración en el usuario
- ☹ Sobrecargamos de trabajo al servidor
- ☹ Sobrecarga en la red

## Reglas de Negocio: Capa de presentación



ALTAS

**Estudiante**

Identificado ¿Violación de clave primaria?

Apellidos ¿Violación de restricción 'not null'?

Nombre

Sexo ¿Valores permitidos?

Fecha de Nacimiento ¿Formato correcto?

Nación

Titulación ¿Clave foránea?

Erasmus ¿Valores permitidos?

AV RN en capa presentación: Claves foráneas

Cliente

ALTAS

Estudiante

Identificador

Apellidos

Nombre

Sexo

Fecha de Nacimiento

Nación

Titulación Erasmus   
Físicas, Licenciatura  
Químicas, Licenciatura

© 2007 J. Lloret, M.A. Zapata y J.C. Ciria <Nº>

AV RN en capa presentación: Valores predefinidos

Cliente

ALTAS

Estudiante

Identificador

Apellidos

Nombre



Sexo  Hombre  Mujer


Fecha de Nacimiento

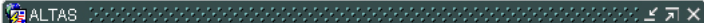
Nación

Titulación Erasmus

© 2007 J. Lloret, M.A. Zapata y J.C. Ciria <Nº>

AV RN en capa presentación: ¿Qué hacer con las fechas?  

Cliente 

ALTAS 


**Estudiante**

Identificador

Apellidos

Nombre

Sexo  Hombre  Mujer



Nacimiento: Día  Mes  Año


Nación


Titulación

Erasmus

© 2007 J. Lloret, M.A. Zapata y J.C. Ciria <Nº>

AV RN en capa presentación: Post-fallo...  

Cliente 

ALTAS 

**Estudiante**

Identificador

Apellidos

Nombre

Sexo  Hombre  Mujer

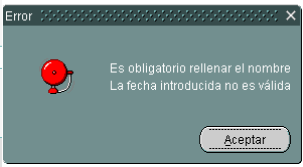
Nacimiento: Día  Mes  Año

Nación

Titulación

Erasmus

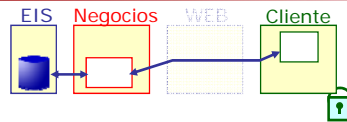
**Error**



Es obligatorio rellenar el nombre  
La fecha introducida no es válida

© 2007 J. Lloret, M.A. Zapata y J.C. Ciria <Nº>

## Reglas de Negocio: ¿Dónde implementarlas?



En la capa de presentación:

FK: listas desplegables

Dominio: botones de radio, casillas de verificación...

- ☺ Amigables e intuitivas
- ☹ No cubren todas las posibilidades
- ☹ Consumen recursos del cliente
- ☹ Por sí solas no garantizan la consistencia de datos en el servidor

© 2007

J. Lloret, M.A. Zapata y J.C. Ciria

<Nº>

## Reglas de Negocio: ¿Dónde implementarlas?




En la capa de negocios

- ☺ Cubren todas las posibilidades
- ☹ Consumen recursos del cliente
- ☹ Por sí solas no garantizan la consistencia de datos en el servidor


© 2007

J. Lloret, M.A. Zapata y J.C. Ciria

<Nº>



## Reglas de Negocio



---


En la capa de negocios...

¿Cómo se implementan en Developer?  
Usando PL/SQL: con funciones, procedimientos y disparadores asociados a eventos y elementos.


¿... y en JDeveloper?

---

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>



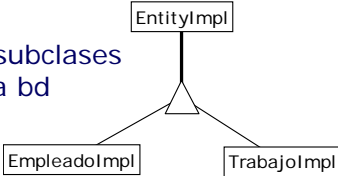
## Reglas de Negocio: JDeveloper



---

ADF BD es un marco (Java Framework Technology): librería de clases orientadas a resolver un problema específico. Usan un fichero de metadatos xml que permite personalización, pero admiten también extensión de clases.

Base Entity Classes:  
Oracle.jbo.server.EntityImpl:  
A partir de ella se extienden las subclases que representan los objetos de la bd



```

classDiagram
    class EntityImpl
    class EmpleadoImpl
    class TrabajoImpl
    EntityImpl <|-- EmpleadoImpl
    EntityImpl <|-- TrabajoImpl
    
```

---

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>

## RN en JDeveloper: Validación a nivel de atributo

Se realiza cada vez que cambia el valor de un atributo.  
Se implementa mediante:

- Reglas de validación: clases Java
- Dominios de validación.
- Método set.

## Validación a nivel de atributo: reglas de validación

Predefinidas:

- **CompareValidator**: comparan un atributo con un literal
- **ListValidator**: comprueban si un atributo está en una lista de literales.
- **RangeValidator**: comprueban si un atributo está comprendido entre dos literales.
- **MethodValidator**: invoca un método en la clase EO.

Custom:

Jdeveloper crea un esqueleto de la clase, con:

- **Boolean validateValue (Object o)** (o es el valor de un atributo)
- **vetoableChange()**: se invoca automáticamente al cambiar el valor de un atributo.
- **Description**: mensaje de información.

## RN en JDeveloper: Validación a nivel de entidad

Afecta a varios atributos.  
Se aplica al hacer commit.  
Se implementa mediante:

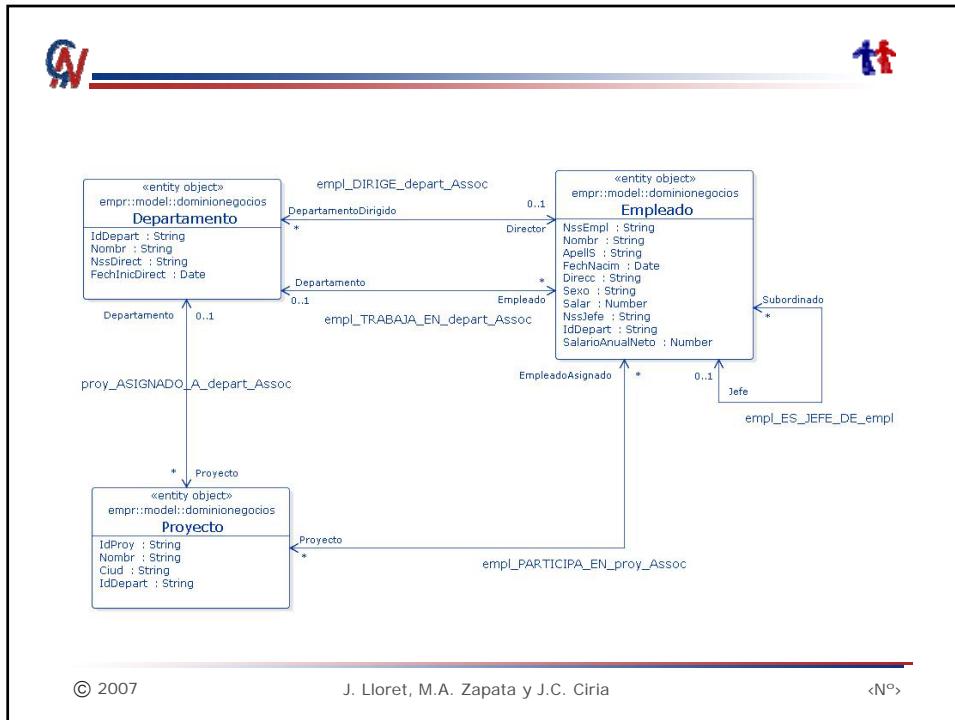
- Reglas de validación a nivel de entidad
- Método `validateEntity()`

## Validación a nivel de entidad: reglas de validación

Predefinidas:

- **MethodValidator**: comparan un atributo con un literal
- **UniqueKeyValidator**: comprueban si un atributo está en una lista de literales.

**Custom**: implementan la interfaz `JbiValidator()`



© 2007

J. Lloret, M.A. Zapata y J.C. Ciria

<Nº>

### RN en capa negocios: Developer

"Ningún trabajador de nuestra empresa cobra menos del salario mínimo interprofesional (570 €)"

PO:

- ⊖ Desparadores
- ⊖ Alertas
- ⊖ Bibliotecas Conectadas
- ⊖ Bloques de Datos
- ⊖ TAB\_EMPL
  - ⊖ Desparadores
  - ⊖ Elementos
    - ⊖ FECH\_NACIM
    - ⊖ DIRECC
    - ⊖ NNS\_EMPL
    - ⊖ NNS\_EMP
    - ⊖ APELL\_S
    - ⊖ SEXO
    - ⊖ SALAR
    - ⊖ Desparadores
      - ⊖ POST-TEXT-ITEM
    - ⊖ NNS\_JEFE
    - ⊖ ID\_DEPART
  - ⊖ Relaciones
- ⊖ LIENZO
  - ⊖ Gráficos
  - ⊖ M\_EMPL
- ⊖ Editores
- ⊖ Listas de Valores
- ⊖ Grupos de Objetos
- ⊖ Parámetros
- ⊖ Menús Emergentes
- ⊖ Unidades de Programa
  - ⊖ LANZA\_MENSAJE (Cuerpo del Procedimiento)
    - ⊖ Clases de Propiedad
    - ⊖ Grupos de Registros
    - ⊖ Informes
    - ⊖ Atributos Visuales
- ⊖ Ventanas
- ⊖ Menús
- ⊖ Bibliotecas de VFN


```
IF (:salar < 570) THEN
    lanza_mensaje ('El salario es menor
        que el interprofesional');
END IF;
```

Módulo creado correctamente.  
Mod: EJEMPLO\_RN Archivo: EJEMPLO\_RN


J. Lloret, M.A. Zapata y J.C. Ciria

<Nº>





## RN en capa negocios: JDeveloper




“Ningún trabajador de nuestra empresa cobra menos del salario mínimo interprofesional (570 €)”


A nivel de atributo

- Reglas de validación
  - Predefinidas:
    - **CompareValidator:**
    - ListValidator:
    - RangeValidator:
    - MethodValidator:
  - Custom:
- Dominios de validación.
- Método set.

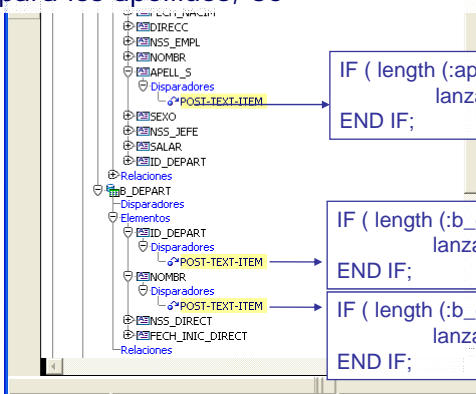
© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>



## RN en capa negocios: JDeveloper



“Las longitudes máximas son:  
para el nombre de un departamento, 20  
para un identificador, 10  
para los apellidos, 30”



```
IF ( length (:apell_s) > 30 ) THEN
    lanza_mensaje ('Apellidos demasiado largos');
END IF;
```

```
IF ( length (:b_departamento.id_depart) > 10 ) THEN
    lanza_mensaje ('Identificador demasiado largo');
END IF;
```

```
IF ( length (:b_departamento.nombr) > 20 ) THEN
    lanza_mensaje ('Identificador demasiado largo');
END IF;
```

Mód: EJEMPLO\_RN Archivo: temáticos\_en\_BD/EJEMPLO\_RN.fmb
C. Ciria
<Nº>



## RN en capa negocios: JDeveloper



“Las longitudes máximas son:  
para el nombre de un departamento, 20  
para un identificador, 10  
para los apellidos, 30”

A nivel de atributo

- Reglas de validación
  - Predefinidas:
    - CompareValidator:
    - ListValidator:
    - RangeValidator:
    - MethodValidator:
  - Custom:
- Dominios de validación.
- Método set.

```
«java class»
empleadoProyecto::client
ValidadorLongitudMáxima
- description : String
- longitudMáxima : int
+ ValidadorLongitudMáxima ()
+ getDescription () : String
+ getLongitudMáxima () : int
+ setDescription (String str) : void
+ setLongitudMáxima (int longitudMáxima) : void
+ validateValue (Object value) : boolean
+ vetoableChange (PropertyChangeEvent eventObj) : void
```

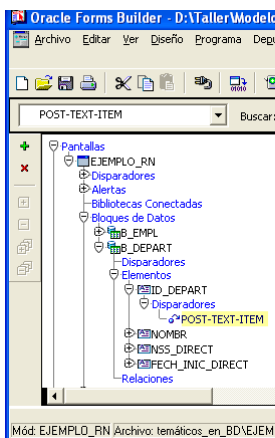
```
description: mensaje de error.
longitudMáxima: adaptable a cada caso.
boolean validateValue (Object value): realiza la validación
public void vetoableChange(PropertyChangeEvent eventObj)
{
  if (!validateValue(eventObj.getNewValue()))
  {
    throw new ValidationException( getDescription() );
  }
}
```



## RN en capa negocios: Developer





“Los identificadores de departamento y proyecto siguen las normas de abreviatura Noesis”



```
DECLARE
  palabra VARCHAR2 (20);
  i INTEGER;
BEGIN
  IF (length(:b_depart.id_depart) > 10) THEN
    lanza_mensaje ('Identificador demasiado largo');
  END IF;

  palabra := :b_depart.id_depart;
  palabra := TRANSLATE ( palabra,
    'aeiouAEIOUáéíóúÁÉÍÓÚ',
    'aaaaaaaaaaaaaaaaaaaaa');

  IF ( INSTR (palabra, 'a',5) != 0) THEN
    lanza_mensaje ('El identificador no cumple la norma Noesis: '||CHR (10)||
    'Hay una vocal más allá de los 4 primeros caracteres');
  END IF;
END;
```


RN en capa negocios: JDeveloper




“Los identificadores de departamento y proyecto siguen las normas de abreviatura Noesis”

A nivel de atributo

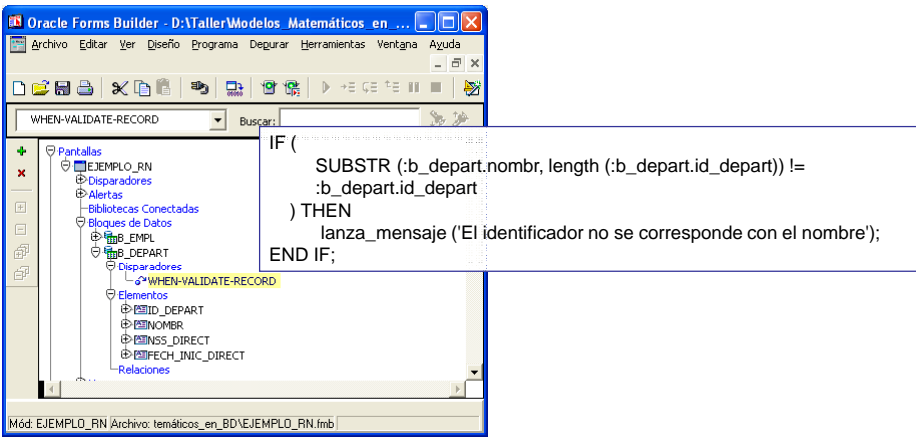
- Reglas de validación
  - Predefinidas:
    - CompareValidator:
    - ListValidator:
    - RangeValidator:
    - MethodValidator:
  - Custom:
- **Dominios de validación.**
- Método set.

```
«java class»
empleadoProyecto::client
DominioAbreviatura
- mData : String
+ equals (Object o) : boolean
+ getMData () : String
+ toString () : String
# validate () : void
```

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>


RN en capa negocios: Developer


“Los identificadores son correctos”



```
IF (
  SUBSTR (:b_depart.nombr, length (:b_depart.id_depart)) !=
  :b_depart.id_depart
) THEN
  lanza_mensaje ('El identificador no se corresponde con el nombre');
END IF;
```



© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>

 RN en capa negocios: JDeveloper 

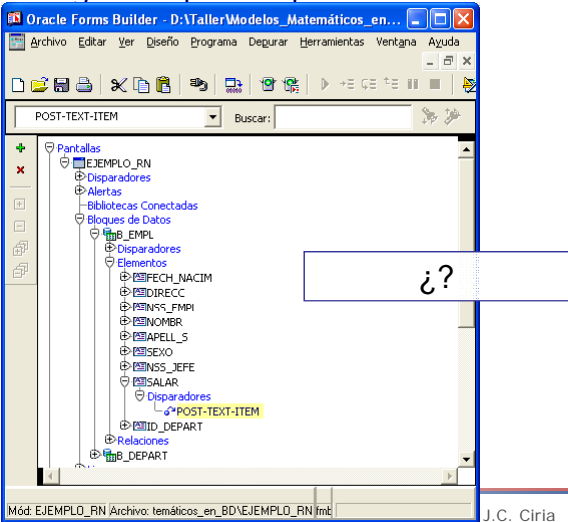
“Los identificadores son correctos”

A nivel de atributo  
A nivel de entidad


© 2007 J. Lloret, M.A. Zapata y J.C. Ciria <Nº>

 RN en capa negocios: Developer 


“Ningún empleado puede cobrar más que su jefe”



Mód: EJEMPLO\_RN Archivo: temáticos\_en\_BD\EJEMPLO\_RN.fmi J.C. Ciria <Nº>



## RN en capa negocios: JDeveloper



“Ningún empleado puede cobrar más que su jefe”

A nivel de atributo

- Reglas de validación
  - Predefinidas:
    - CompareValidator:
    - ListValidator:
    - RangeValidator:
    - MethodValidator:
  - Custom:
- Dominios de validación.
- **Método set**

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>




## RN en capa negocios: Developer




“Los empleados sin jefe, y sólo ellos, cobran más de 8000 € al mes”



© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>



## RN en capa negocios: JDeveloper




“Los empleados sin jefe, y sólo ellos, cobran más de 8000 € al mes”


A nivel de atributo  
**A nivel de entidad**







```
protected void validateEntity()
{
    super.validateEntity();
    String jefe = getNssJefe();
    if ( (jefe == null)
        && (getSalar().floatValue() < 8000 )
    )
        throw new oracle.jbo.JboException ("Este jefe gana muy poco");
    else
    {
        throw new oracle.jbo.JboException ("Este gana como si fuera jefe, y no lo es");
    }
}
}
```

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>



## Reglas de Negocio



	Regla de validación	Dominio	Set
Reusabilidad	  puede aplicarse a atributos de distintos tipos		No
Facilidad de programación			
¿Se requiere acceder a otros atributos?			 usando get
Estilo de programación preferido			 Estándar Java

© 2007
J. Lloret, M.A. Zapata y J.C. Ciria
<Nº>