

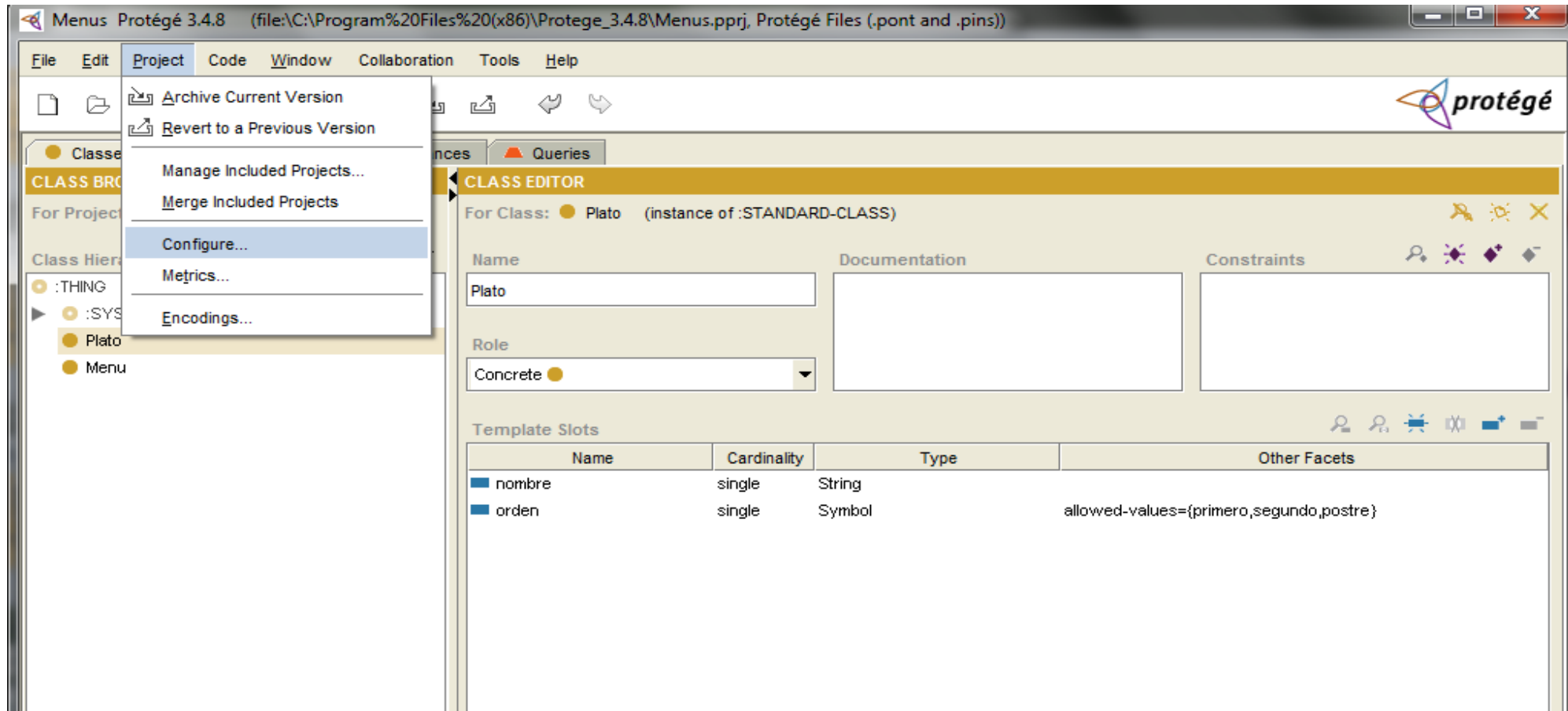
# JessTab: Protégé con Jess

- JessTab es un puente entre Protégé y Jess
- Proporciona una consola de Jess en una pestaña de Protégé
- Permite incluir reglas Jess en las ontologías de Protégé
- Las reglas creadas pasan a formar parte de la ontología
- Se pueden crear correspondencias (mappings) entre instancias de la ontología y hechos en Jess
- Es posible manejar con Jess las ontologías y bases de conocimiento desarrolladas con Protégé
- Jess y Protégé pueden ser utilizados desde Java

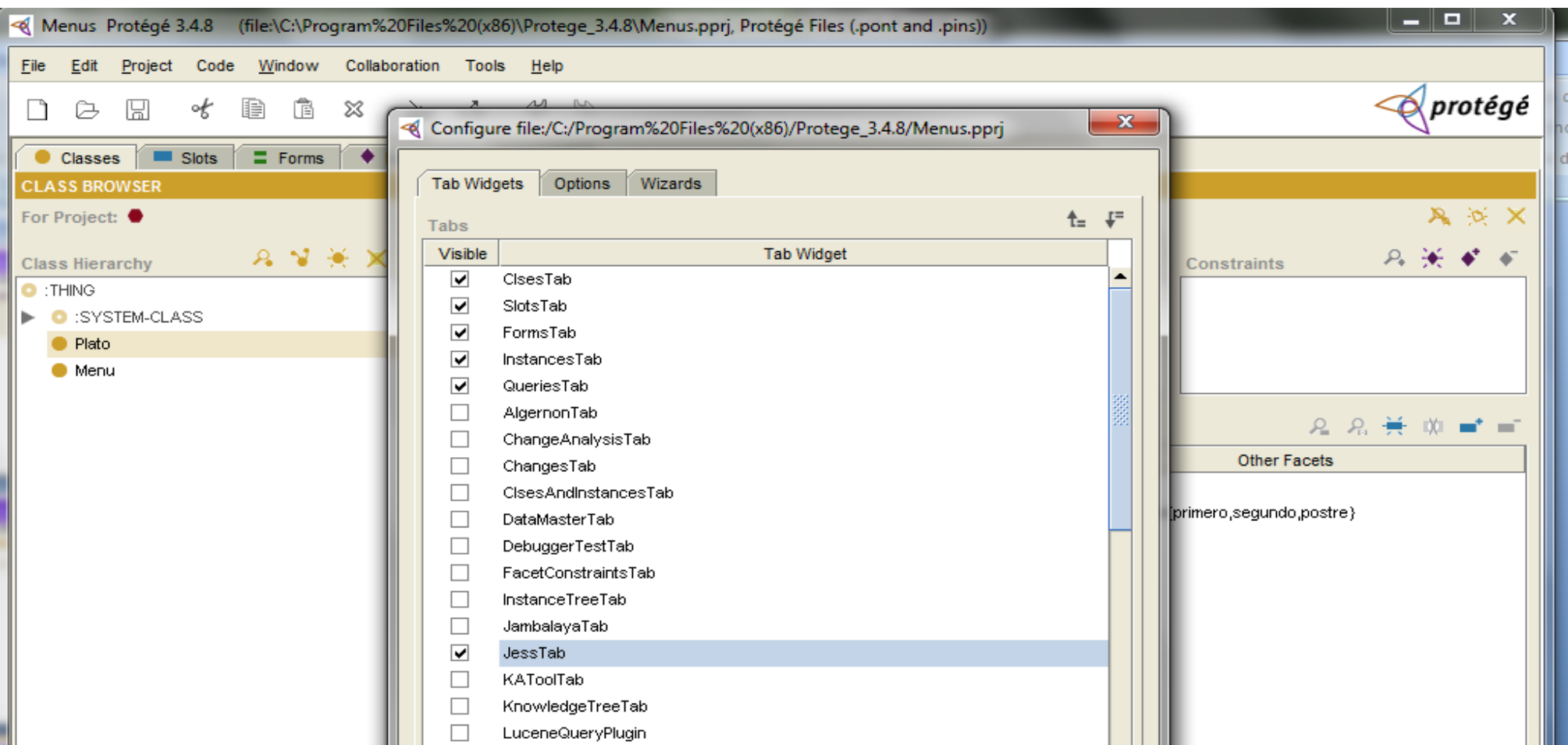
## Utilización de JessTab

- Previamente debe estar instalado Protégé 3.4.8 que lleva incluido JessTab
- Copiar el archivo jess.jar (incluido en la distribución de Jess)
- Pegarlo en el subdirectorio /plugins/se.liu.ida.JessTab de la instalación de Protégé 3.4.8. Este subdirectorio contiene ya otros dos archivos: JessTab17.jar y plugin.properties.

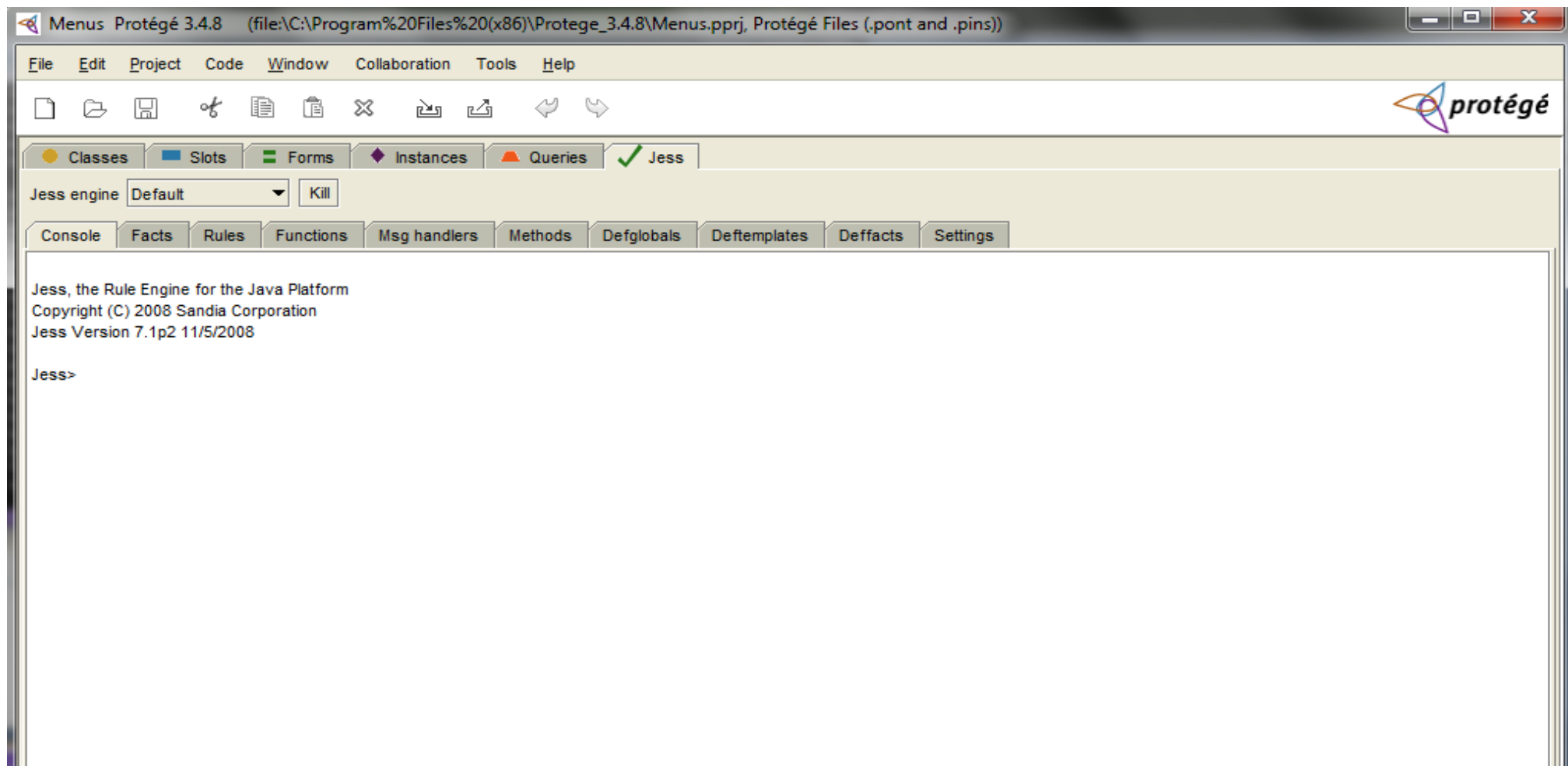
# Configurar Protégé para incluir Jess



# Configurar Protégé para incluir Jess



# Consola de Jess en Protégé



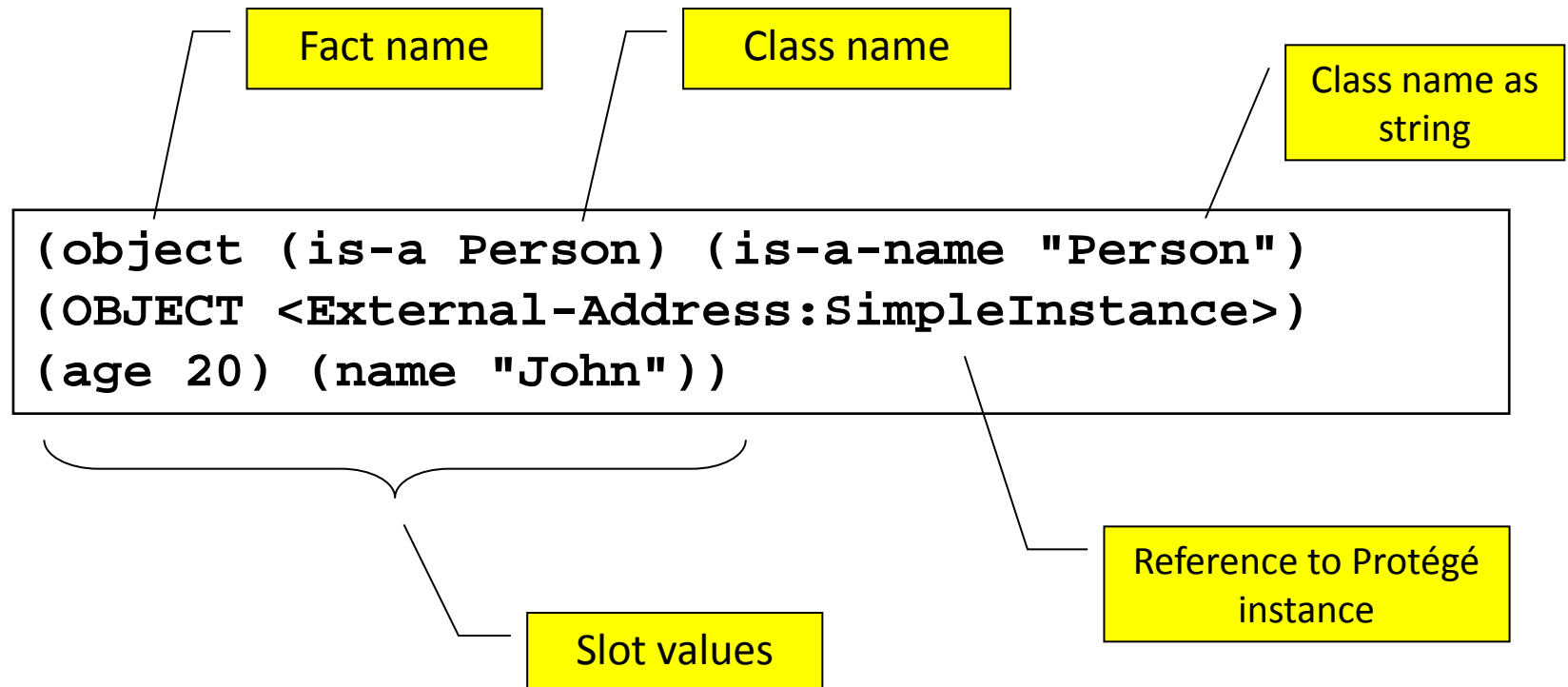
# Ejecución de un programa Jess



El programa en Jess se puede ejecutar en *batch* o pegar su contenido en el panel inferior y pulsar *enter* o ejecutar instrucción a instrucción pulsando *enter* cada vez

# Correspondencia entre Jess y Protégé

- Hecho Jess correspondiente a una instancia de la clase Persona creada en Protégé



## Ejemplo de uso de JessTab

- A partir de una sencilla base de conocimiento de platos (creada en Protégé), con dos instancias de primer plato, dos de segundo plato y dos de postre, generar con Jess instancias de la clase Menú también definida en la misma ontología Protégé.
- En este ejemplo, todos los datos se han metido en Protégé y usamos una regla Jess para generar resultados en Protégé.



# Clases en la ontología Protégé

The screenshot displays the Protégé 3.4.8 application window. The title bar indicates the file path: (file:\C:\Program%20Files%20(x86)\Protege\_3.4.8\Menus.pprj, Protégé Files (.pont and .pins)). The menu bar includes File, Edit, Project, Code, Window, Collaboration, Tools, and Help. The toolbar contains icons for file operations and editing. The main interface is divided into two panes: the CLASS BROWSER on the left and the CLASS EDITOR on the right.

**CLASS BROWSER**

For Project: ●

Class Hierarchy

- :THING
  - :SYSTEM-CLASS
    - Plato
    - Menu

**CLASS EDITOR**

For Class: ● Plato (instance of :STANDARD-CLASS)

Name: Plato

Documentation: [Empty text area]

Constraints: [Empty text area]

Role: Concrete ●

Template Slots

| Name   | Cardinality | Type   | Other Facets                            |
|--------|-------------|--------|---|
| nombre | single      | String |   |
| orden  | single      | Symbol | allowed-values={primero,segundo,postre} |

# Clase Menú

Menus Protégé 3.4.8 (file:\C:\Program%20Files%20(x86)\Protege\_3.4.8\Menus.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes Slots Forms Instances Queries Jess

**CLASS BROWSER**

For Project: Menus

Class Hierarchy

- :THING
  - :SYSTEM-CLASS
    - Menu
    - Plato

**CLASS EDITOR**

For Class: Menu (instance of :STANDARD-CLASS)

Name

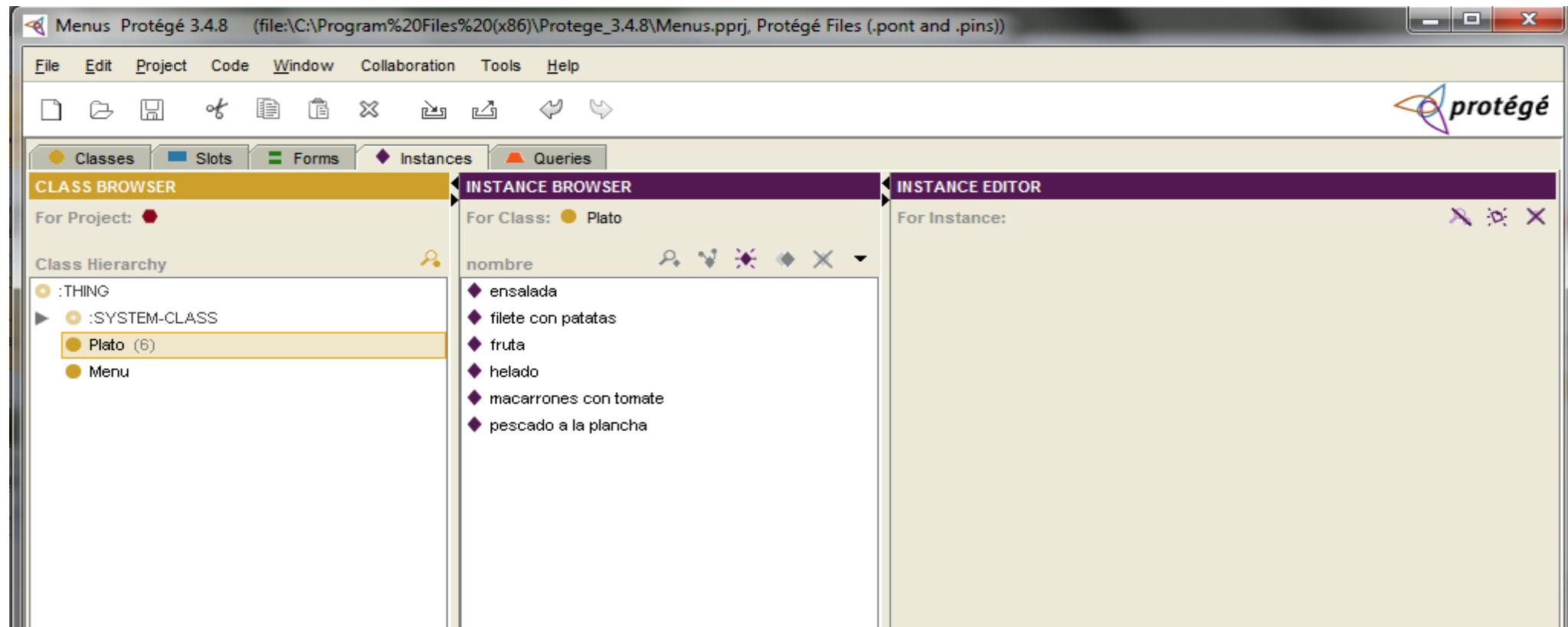
Documentation

Role

Template Slots

| Name    | Cardinality | Type              |
|---------|-------------|-------------------|
| postre  | single      | Instance of Plato |
| primero | single      | Instance of Plato |
| segundo | single      | Instance of Plato |

# Instancias de Plato



# Programa en Jess para el ejemplo de Menús

; programa en Jess con Protege

(mapclass Plato)

(mapclass Menu)

Establece la correspondencia entre Protégé y Jess

(defrule genera "Coge instancias de la clase Plato, genera instancias de Menú y las guarda en la ontología"

?h1 <- (object (is-a Plato)(orden primero))

Coge una instancia de cada plato en Protégé

?h2 <- (object (is-a Plato)(orden segundo))

?h3 <- (object (is-a Plato)(orden postre))

=>

(make-instance of Menu (primero ?h1) (segundo ?h2) (postre ?h3))

)

(reset)

(run)

(facts)

Construye instancias de Menú en Protégé, con todas las combinaciones posibles de instancias de cada plato

# Programa en Jess para el ejemplo de Menús

Alternativamente, en vez de punteros a los hechos instancia, se pueden usar las referencias externas

```
; programa en Jess con Protege
```

```
(mapclass Plato)
```

```
(mapclass Menu)
```

```
(defrule genera "Coge instancias de la clase Plato, genera instancias de Menú y  
las guarda en la ontología"
```

```
(object (is-a Plato) (OBJECT ?h1) (orden primero))
```

```
(object (is-a Plato) (OBJECT ?h2) (orden segundo))
```

```
(object (is-a Plato) (OBJECT ?h3) (orden postre))
```

```
=>
```

```
(make-instance of Menu (primero ?h1) (segundo ?h2) (postre ?h3))
```

```
)
```

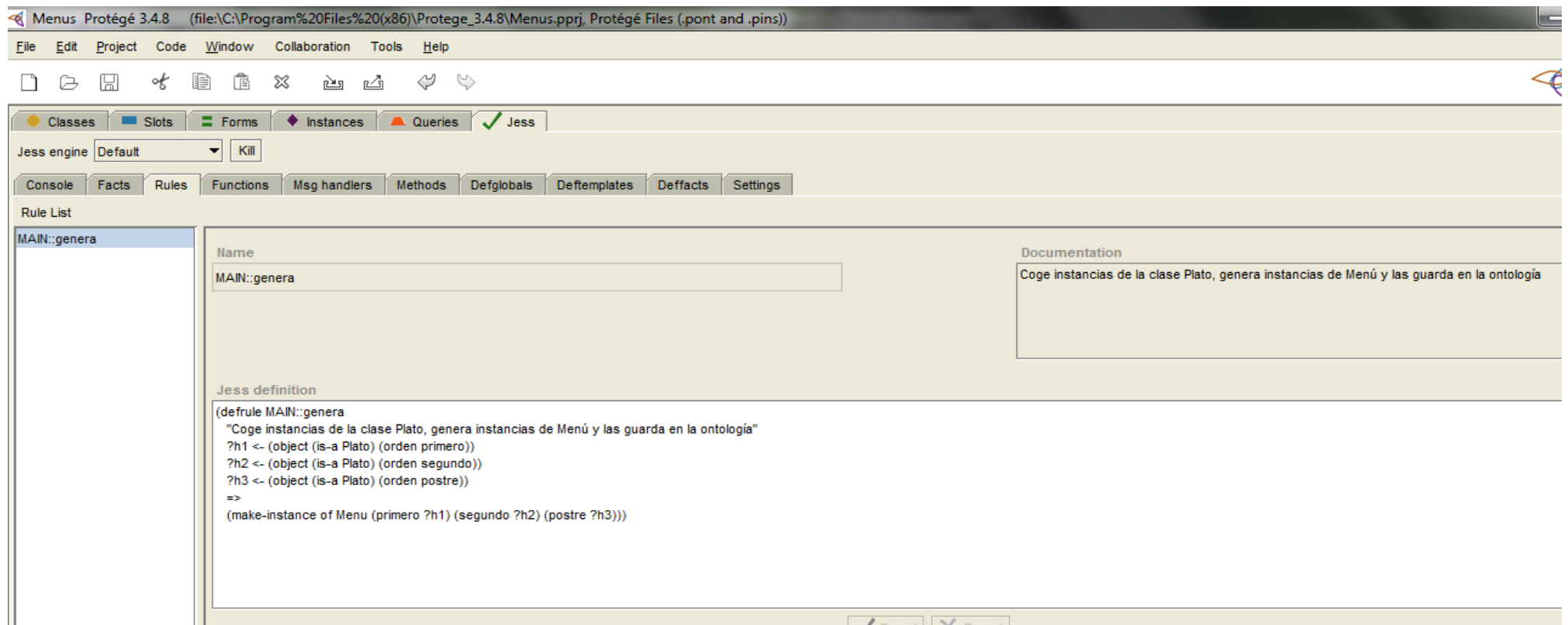
```
(reset)
```

```
(run)
```

```
(facts)
```

# Regla incorporada a la ontología

Las reglas introducidas en el programa Jess se pueden visualizar desde la pestaña Rules y guardarlas junto con la ontología.



# Hechos Jess correspondientes a la ontología

Menus Protégé 3.4.8 (file:\C:\Program%20Files%20(x86)\Protege\_3.4.8\Menus.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes Slots Forms Instances Queries Jess

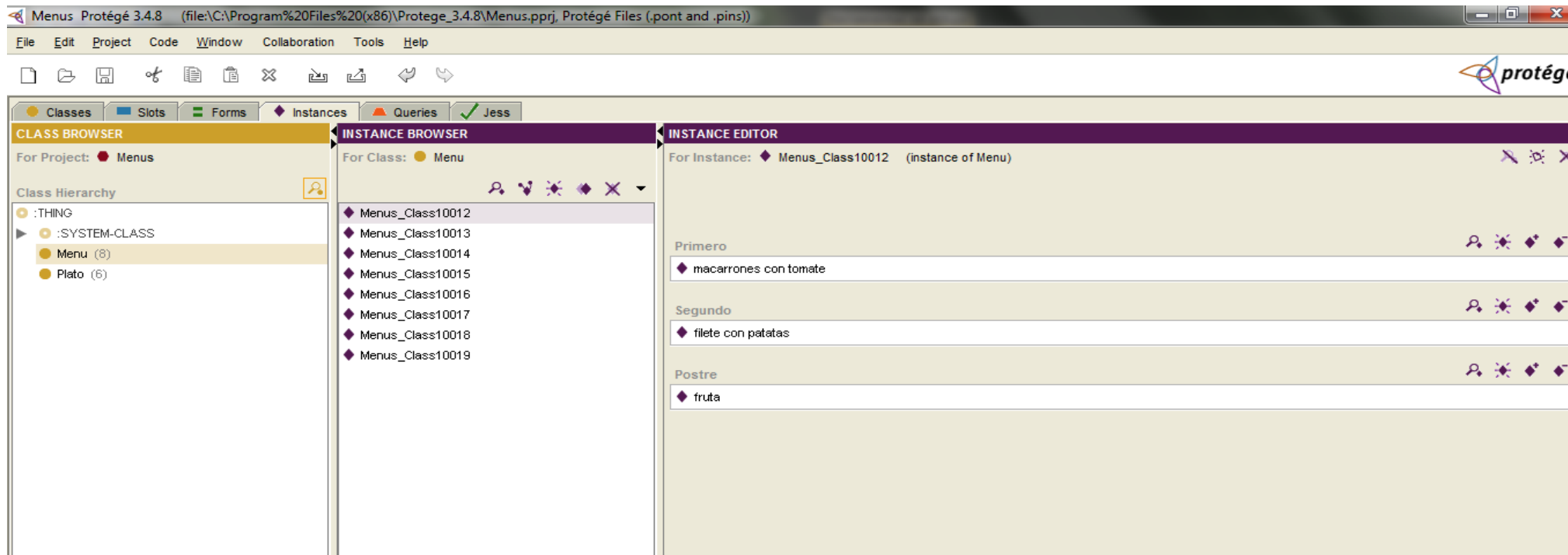
Jess engine Default Kill

Console Facts Rules Functions Msg handlers Methods Defglobals Deftemplates Deffacts Settings

Fact List

| Id | Fact   |
|----|--|
| 0  | (MAIN::initial-fact)   |
| 1  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "helado") (orden postre))   |
| 2  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "fruta") (orden postre))  |
| 3  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "pescado a la plancha") (orden segundo))  |
| 4  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "filete con patatas") (orden segundo))  |
| 5  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "ensalada") (orden primero))  |
| 6  | (MAIN::object (is-a Plato) (is-a-name "Plato") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (nombre "macarrones con tomate") (orden primero))   |
| 10 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 14 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 18 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 22 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 26 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 30 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 34 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |
| 38 | (MAIN::object (is-a Menu) (is-a-name "Menu") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (segundo <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (postre <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>)) |

# Menús generados





# Recomendaciones

- Empezar con ejemplos sencillos
- Usar (`mapclass NombreClase`) para establecer correspondencias con las clases definidas en la ontología
- Usar (`facts`) para ver los hechos Jess correspondientes a la ontología
- Definir reglas cuyos patrones encajen en esos hechos
- Ejecutar las reglas con (`run`)

# Ejemplo de carga de datos de móviles desde Jess

Móviles Protégé 3.4.8 (file:\C:\Program%20Files%20(x86)\Protege\_3.4.8\Móviles.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes Slots Forms Instances Queries

CLASS BROWSER

For Project:

Class Hierarchy

- :THING
- :SYSTEM-CLASS
- MóvilP

CLASS EDITOR

For Class: MóvilP (instance of :STANDARD-CLASS)

Name

MóvilP

Documentation

Role

Concrete

Template Slots

| Name   | Cardinality | Type    |
|--------|-------------|---------|
| marca  | single      | String  |
| modelo | single      | String  |
| precio | single      | Integer |

# Ejemplo de carga de datos de móviles desde Jess

## Programa en Jess

```
(mapclass MovilP)
```

Clase en Protégé

```
(deftemplate MovilJ
```

Template en Jess

Podrían llamarse igual

```
  (slot marca)
  (slot modelo)
  (slot precio))
```

```
(deffacts ini
```

```
  (MovilJ (marca Samsung)(modelo "Galaxy S4") (precio 699))
  (MovilJ (marca Apple) (modelo "iPhone 5") (precio 660)))
```

```
(defrule cargar      (MovilJ (marca ?m)(modelo ?mo) (precio ?p))
```

```
=>
```

```
  (make-instance of MovilP (marca ?m)(modelo ?mo) (precio ?p)))
```

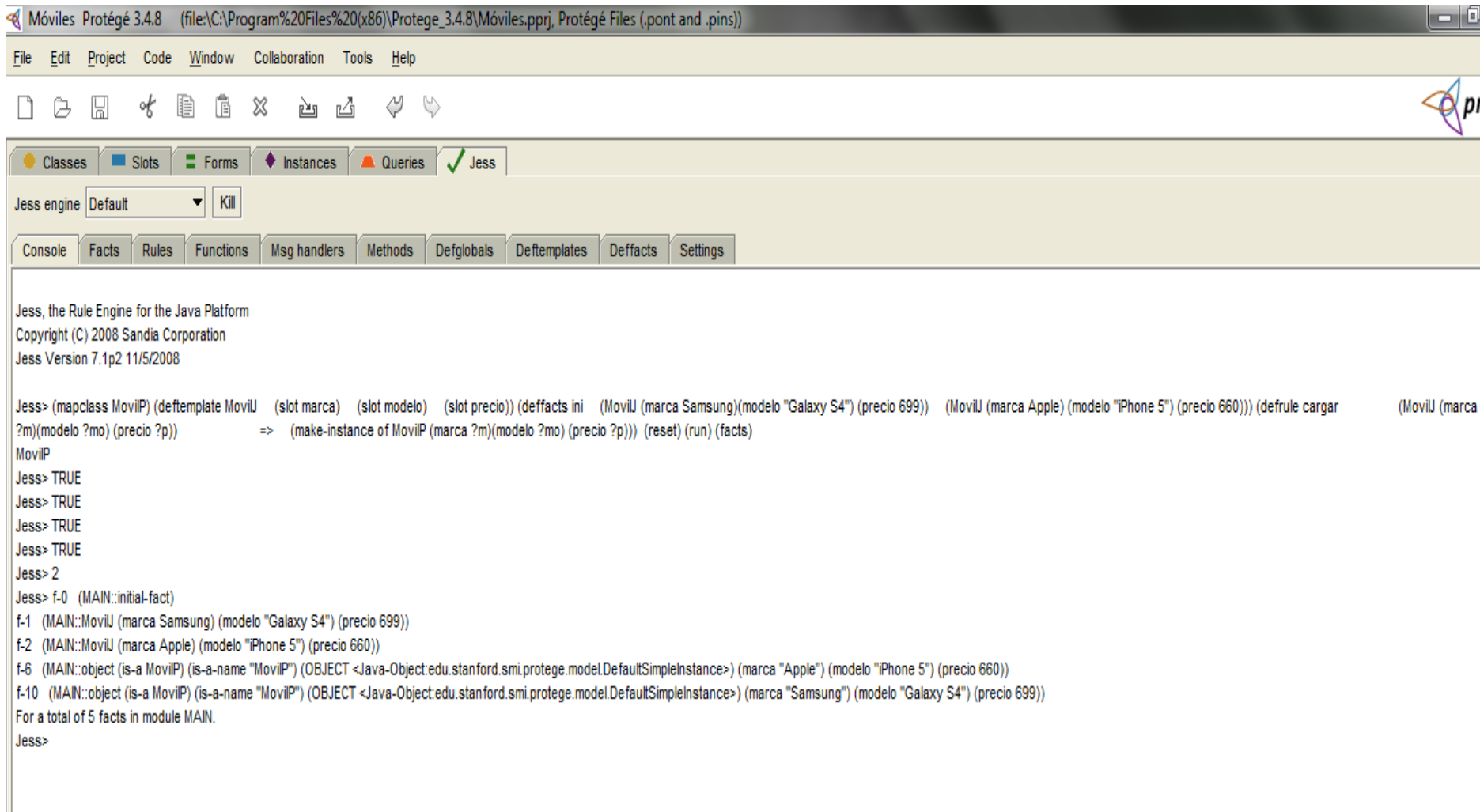
```
(reset)
```

```
(run)
```

```
(facts)
```

# Ejemplo de carga de datos de móviles desde Jess

## Programa Jess cargado y ejecutado



Móviles Protégé 3.4.8 (file:C:\Program%20Files%20(x86)\Protege\_3.4.8\Móviles.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes Slots Forms Instances Queries Jess

Jess engine Default Kill

Console Facts Rules Functions Msg handlers Methods Defglobals Deftemplates Deffacts Settings

Jess, the Rule Engine for the Java Platform  
Copyright (C) 2008 Sandia Corporation  
Jess Version 7.1p2 11/5/2008

```
Jess> (mapclass MovilP) (deftemplate MovilU (slot marca) (slot modelo) (slot precio)) (def facts ini (MovilU (marca Samsung)(modelo "Galaxy S4") (precio 699)) (MovilU (marca Apple) (modelo "iPhone 5") (precio 660))) (defrule cargar (MovilU (marca ?m)(modelo ?mo) (precio ?p)) => (make-instance of MovilP (marca ?m)(modelo ?mo) (precio ?p))) (reset) (run) (facts)
MovilP
Jess> TRUE
Jess> TRUE
Jess> TRUE
Jess> TRUE
Jess> 2
Jess> f-0 (MAIN::initial-fact)
f-1 (MAIN::MovilU (marca Samsung) (modelo "Galaxy S4") (precio 699))
f-2 (MAIN::MovilU (marca Apple) (modelo "iPhone 5") (precio 660))
f-6 (MAIN::object (is-a MovilP) (is-a-name "MovilP") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (marca "Apple") (modelo "iPhone 5") (precio 660))
f-10 (MAIN::object (is-a MovilP) (is-a-name "MovilP") (OBJECT <Java-Object:edu.stanford.smi.protege.model.DefaultSimpleInstance>) (marca "Samsung") (modelo "Galaxy S4") (precio 699))
For a total of 5 facts in module MAIN.
Jess>
```

# Ejemplo de carga de datos de móviles desde Jess

## Se han añadido 2 instancias en Protégé

The screenshot shows the Protégé 3.4.8 interface with the following components:

- Top Menu Bar:** File, Edit, Project, Code, Window, Collaboration, Tools, Help.
- Top Toolbar:** Icons for file operations (new, open, save, delete, copy, paste, undo, redo) and navigation.
- Left Pane (CLASS BROWSER):**
  - For Project:
  - Class Hierarchy:
    - :THING
    - :SYSTEM-CLASS
    - MovilP (2)
- Middle Pane (INSTANCE BROWSER):**
  - For Class: MovilP
  - modelo:
    - iPhone 5
    - Galaxy S4
- Right Pane (INSTANCE EDITOR):**
  - For Instance: iPhone 5 (instance of MovilP, internal name is Móviles\_Class8)
  - Marca:
  - Modelo:
  - Precio:

# Herencia múltiple

- prueba1 es una instancia de la clase Movil1.
- para hacer que también lo sea de la clase Movil2, la añadimos a la lista de instancias directas de la clase Movil2

```
(defrule MAIN::R
```

```
  (object (is-a Movil1) (OBJECT ?h) (modelo "prueba1"))
```

```
  (object (is-a :STANDARD-CLASS) (:NAME "Movil2")
          (:DIRECT-INSTANCES $?x))
```

```
=>
```

```
  (slot-set "Movil2" :DIRECT-INSTANCES
            (insert$ ?x (+ 1 (length$ ?x)) ?h))
```

```
)
```

# Funciones Jess para manejar bases de conocimiento

|                     |                      |                        |
|---------------------|----------------------|------------------------|
| mapclass            | slot-range           | instancep              |
| mapinstance         | slot-allowed-values  | instance-existp        |
| unmapinstance       | slot-allowed-classes | instance-name          |
| defclass            | slot-allowed-parents | instance-address       |
| make-instance       | slot-documentation   | instance-addressp      |
| initialize-instance | slot-sources         | instance-namep         |
| modify-instance     | facet-get            | slot-existp            |
| duplicate-instance  | facet-set            | slot-default-value     |
| definstances        | class                | set-kb-save            |
| unmake-instance     | class-existp         | get-kb-save            |
| slot-get            | class-abstractp      | load-kb-definitions    |
| slot-set            | class-reactivep      | load-project           |
| slot-replace\$      | superclassp          | include-project        |
| slot-insert\$       | subclassp            | save-project           |
| slot-delete\$       | class-superclasses   | jesstab-version-number |
| slot-facets         | class-subclasses     | jesstab-version-string |
| slot-types          | get-defclass-list    | get-knowledge-base     |
| slot-cardinality    | class-slots          | get-tabs               |

# Referencias

- Manual de JessTab:  
<http://www.ida.liu.se/~her/JessTab/JessTab.pdf>
- Tutorial de JessTab:  
<http://www.ida.liu.se/~her/JessTab/tutorial06/>
- Ejemplo de programa en Jess que usa una ontología Protégé:  
<http://www.ida.liu.se/~her/JessTab/tutorial06/newspaper.jess>
- Transparencias del tutorial:  
<http://www.ida.liu.se/~her/JessTab/tutorial06/JessTabTutorial.ppt>