

# Ingeniería de ontologías

## Metodologías de diseño - Introducción

# ¿Por qué ontologías?

Ontología	MER, EER
<b>Modelo conceptual</b> para un dominio	<b>Modelo conceptual</b> para una aplicación
Reusable por varias aplicaciones	
Vocabulario consensuado, estándar	Conceptos, entidades de la aplicación
Formalizadas en lenguaje basado en lógica	En general, representación gráfica
Independiente de la implementación	Independiente de la implementación



Ontología	Modelo relacional
<b>Base de conocimiento</b>	<b>Base de conocimiento</b>
Reglas basadas en lógica como parte del modelo (expresividad, nivel de abstracción)	Restricciones de integridad como parte del modelo
Razonamiento: detección de inconsistencias e inferencia de conocimiento implícito	Verificación de restricciones de integridad
Mundo abierto (extensiones mundo cerrado)	Mundo cerrado
+ semántica   - desempeño computacional	- semántica   + desempeño computacional

# Clasificación de ontologías

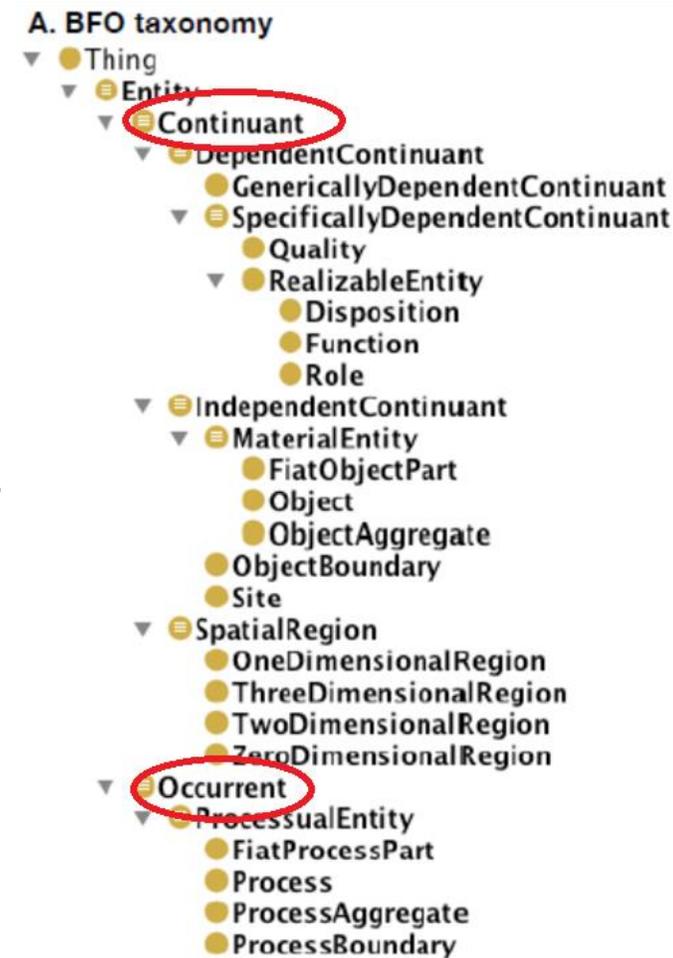
## Fundacionales

- Categorías y relaciones que describen el universo desde un punto de vista filosófico

**Basic Formal Ontology, Unified Foundational Ontology**

## De dominio

- Describen los términos de un dominio (de referencia)  
SNOMED CT, CIDOC-CRM
- Ontologías de aplicación: restricciones y relaciones más ricas  
Food ontology



# Metodologías de diseño de ontologías

Metodologías que abordan todo el ciclo de desarrollo.

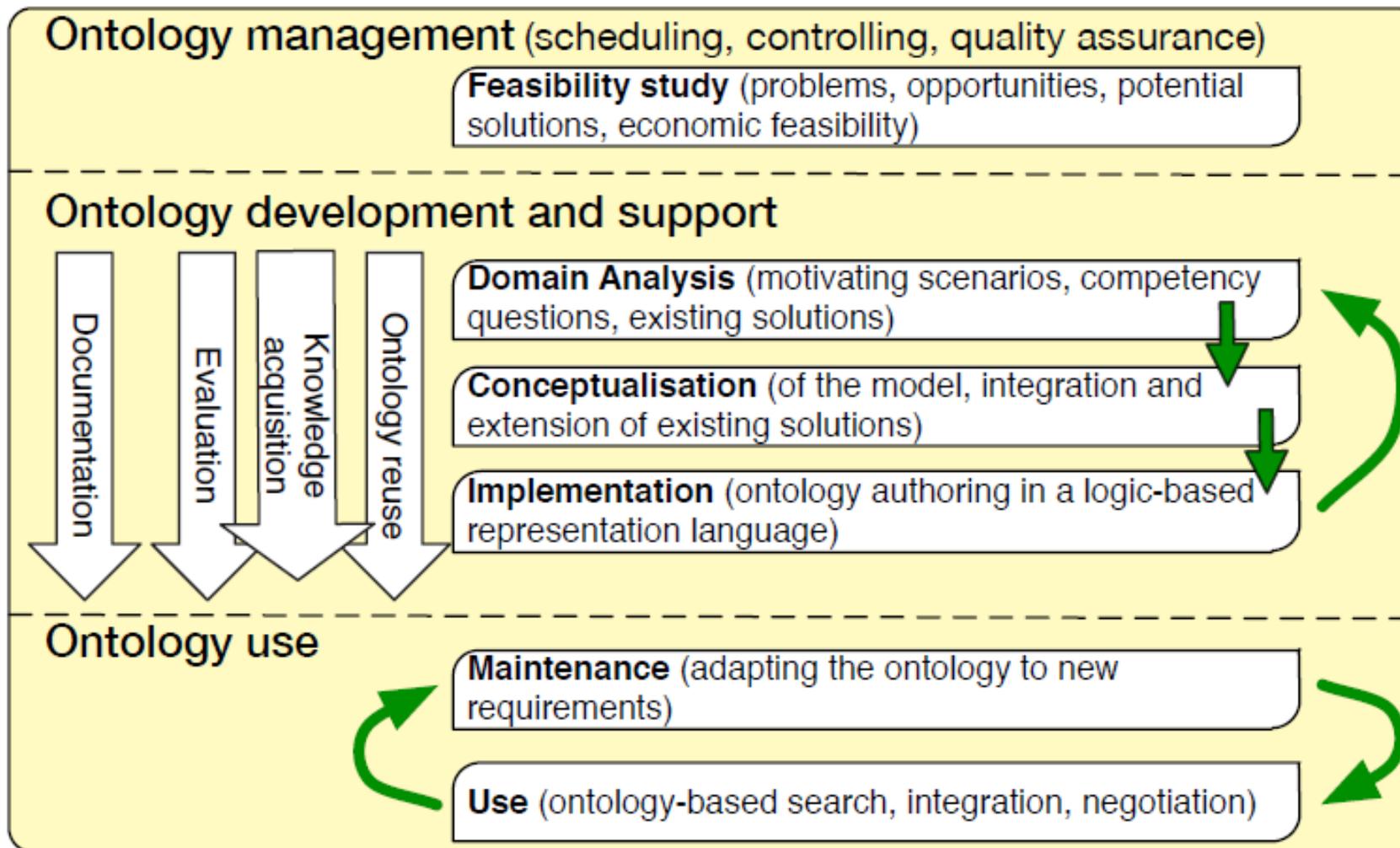
- Describen actividades a alto nivel, con algunas guías de cómo abordarlas
- Foco en el proceso de desarrollo
- Cómo organizar el trabajo del equipo de desarrollo
- Trabajo equipo reducido vs trabajo colaborativo
- Desarrollo de ontologías monolíticas vs redes de ontologías

# Metodologías de diseño de ontologías

Métodos, guías y buenas prácticas que apuntan a mejorar la calidad de las ontologías

- Profundizan más en la construcción de ontologías de calidad
- Enfoques de desarrollo: top-down, bottom-up, patrones (y antipatrones)
- Consistencia lógica (razonador) vs consistencia ontológica (conceptual)
- Buenas prácticas y decisiones de diseño, formalización, algunos Tips
- ¿Cómo empezar a construir un modelo? Desde cero o reuso
- Ontologías como componentes de una aplicación

# Metodologías que abordan todo el ciclo de desarrollo



Elena Simperl, Malgorzata Mochol, and Tobias Bürger. *Achieving maturity: the state of practice in ontology engineering in 2009*

# Metodologías que abordan todo el ciclo de desarrollo

Methontology

NeOn

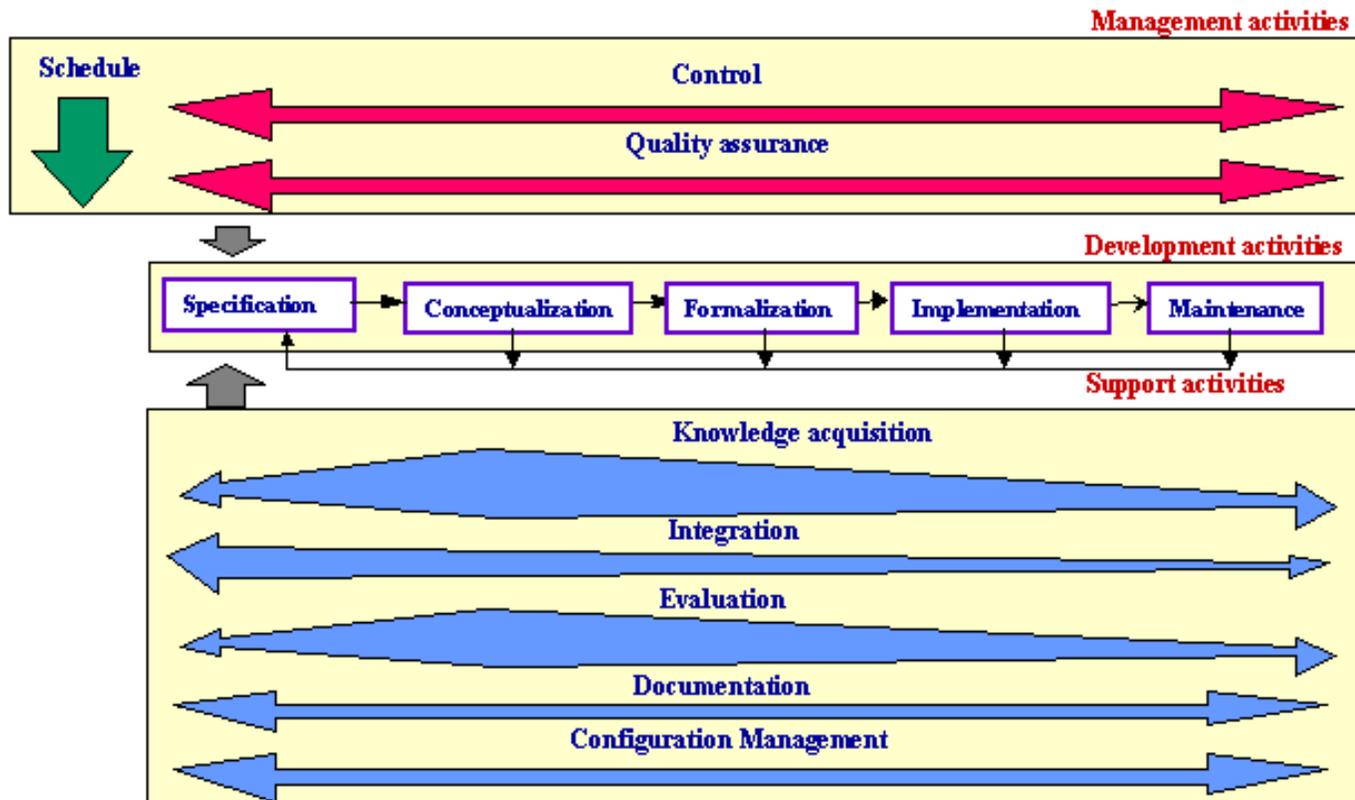
Ontology Summit Communique

Metodologías ágiles:

- XD: eXtremeDesign
- SAMOD

# Methontology

- Pensada para construir una ontología monolítica, en los 90 ´s
- Un equipo reducido de ingenieros de ontologías, en un mismo lugar de trabajo
- Ciclo de vida: iterativo, prototipos evolutivos



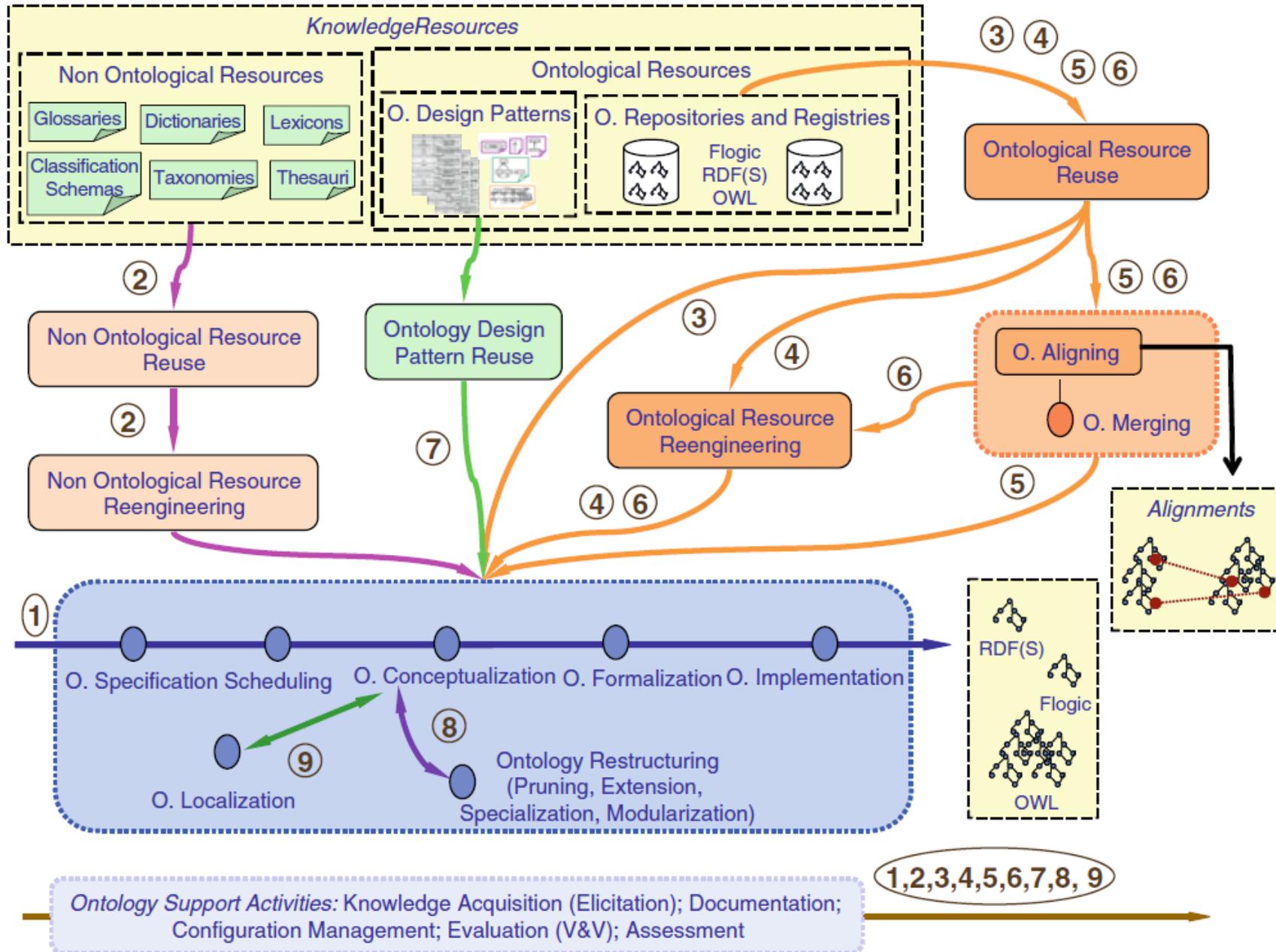
Si bien ha sido reemplazada por otras metodologías, **Methontology es la base:** los procesos de gestión, desarrollo y soporte se mantienen

Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho. Ontological Engineering.

# NeOn

- Pensada para construir **redes de ontologías**
- **Varios equipos** (posiblemente en diferentes lugares) trabajan en forma **colaborativa**
- Ciclos de vida: cascada, iterativo incremental (dependiendo del proyecto)
- Incluye las actividades de Mehontology, agrega **nuevas actividades** (reúso de ontologías, módulos, patrones de diseño, etc.)
- **9 escenarios** que combinan las actividades de diferente forma:
  1. *Desde cero: desde la especificación a la implementación*
  2. *Reúso y reingeniería de recursos no ontológicos*
  3. *Reúso de recursos ontológicos*
  4. *Reúso y reingeniería de recursos ontológicos*
  5. *Reúso y merge de recursos ontológicos*
  6. *Reúso, merge y reingeniería de recursos ontológicos*
  7. *Reúso de patrones de diseño*
  8. *Reestructura de recursos ontológicos*
  9. *Adaptación de ontologías a otros idiomas (localization)*

# NeOn



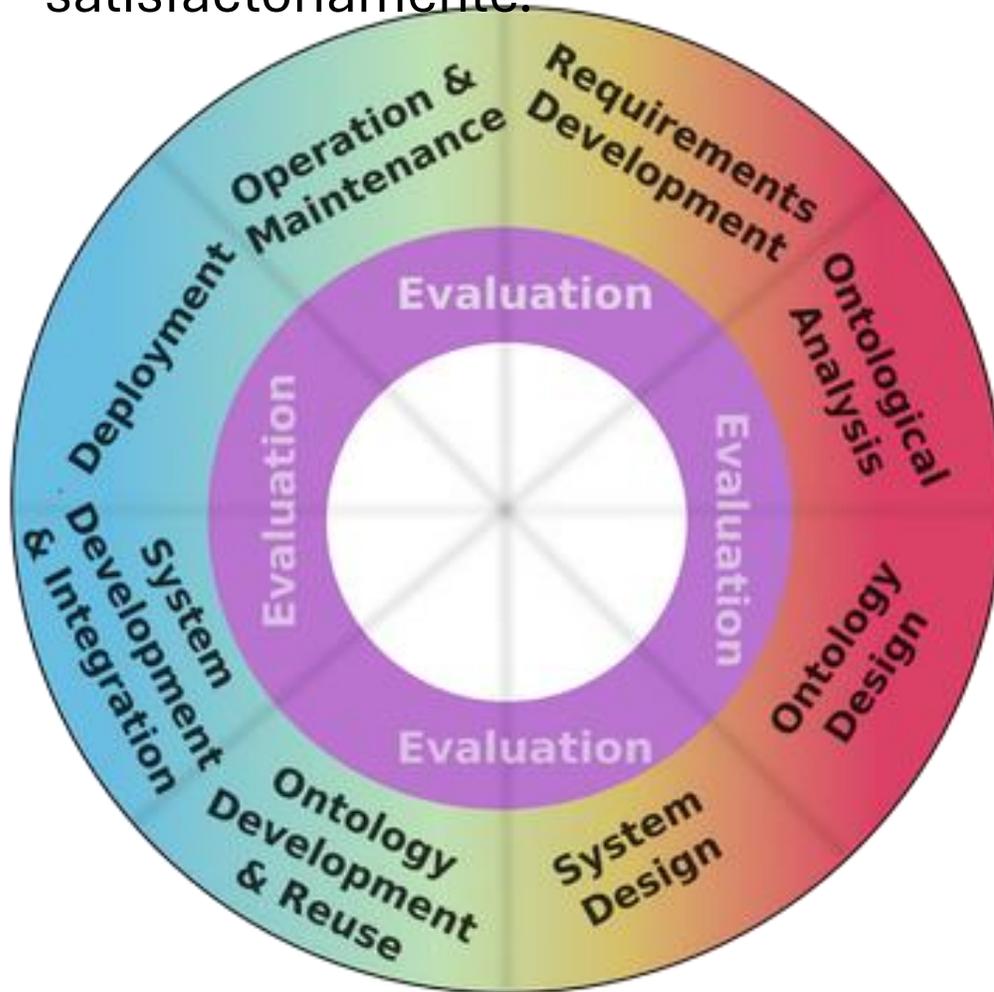
Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, Aldo Gangemi. *Ontology Engineering in a Networked World.*

# Ontology Summit Communique

- Evaluación en todo el ciclo de vida
- Un modelo de ciclo de vida y un criterio de evaluación para cada fase
- Se evalúan 5 aspectos de calidad:
  - Los humanos comprenden correctamente la ontología? *Inteligibilidad*
  - La ontología representa con exactitud el dominio? *Fidelidad*
  - Las decisiones de diseño de la ontología son consistentes? *Destreza (calidad de diseño)*
  - La representación del dominio se ajusta a los requerimientos para el uso previsto?  
*Aptitud*
  - La ontología implementada cumple con los requerimientos del sistema de información del que forma parte? *Capacidad de implementación*
- Trazabilidad con requerimientos, evaluación con respecto al cumplimiento de los requerimientos

# Ontology Summit Communique

En **cada fase** se plantea un conjunto de **preguntas que deben ser respondidas** satisfactoriamente.



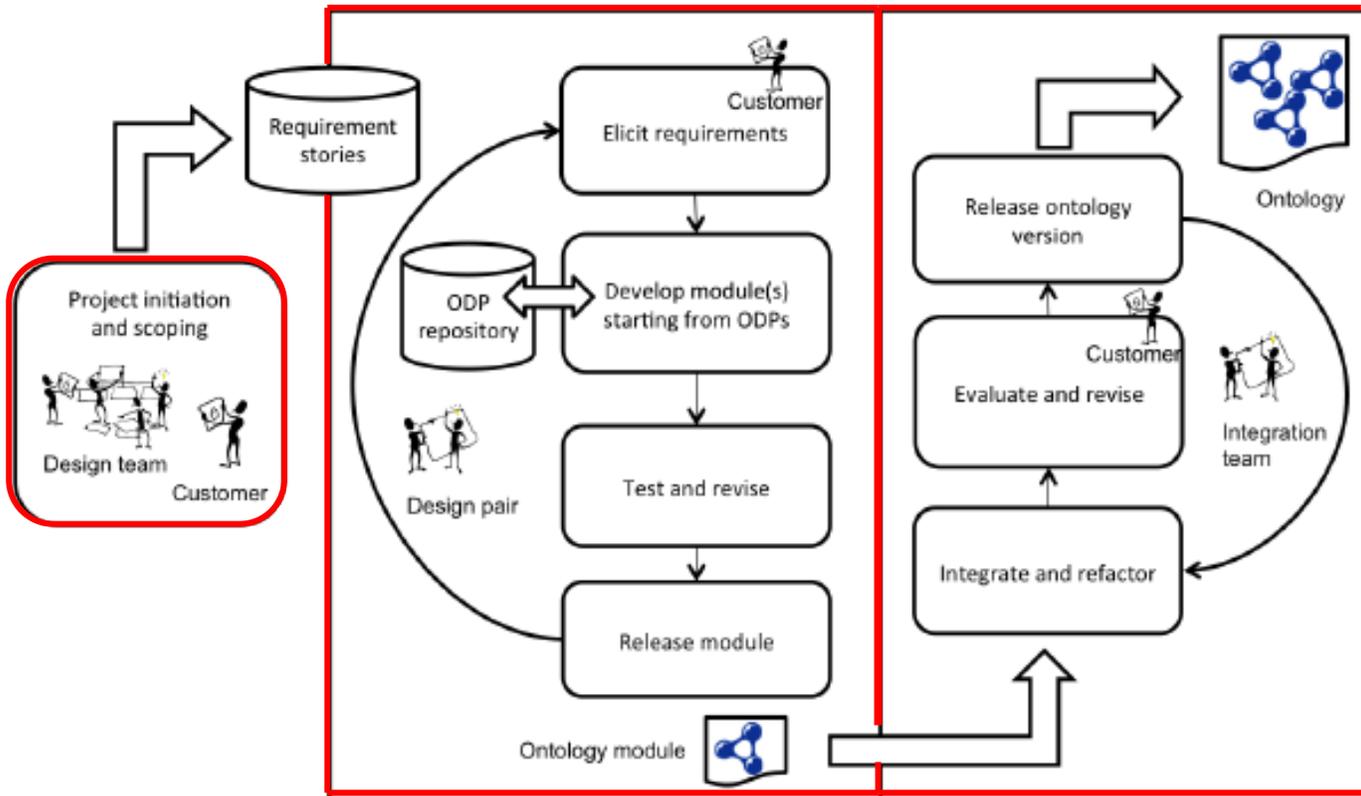
## Fase de requerimientos

- Por qué se necesita esta ontología? (Cuál es el fundamento? Cuáles son los beneficios esperados)?
- Existen ontologías o estándares que deban ser reutilizados?
- Cuáles son las preguntas de competencia? (Qué preguntas debería responder la ontología?)

## Fase de diseño

- Qué operaciones son resueltas por los componentes del sistema, asistidos por la ontología? De qué forma los requerimientos del negocio se cumplen con estas operaciones y componentes específicos?
- Cómo se construirá, evaluará y mantendrá la ontología? Qué herramientas se necesitan para el desarrollo, la evaluación, la gestión de la configuración y el mantenimiento de la ontología?

# eXtreme Design (XD)



E. Blomqvist, K. Hammar, and Valentina Presuttii. Engineering ontologies with patterns - the extreme design methodology

## Ciclo de vida ágil e iterativo

### Fase 1: *Iniciación y alcance del Proyecto*

- Se recolectan los requerimientos en forma de historias de usuarios

### Fase 2: *Fase iterativa de producción de nuevos módulos de la red de ontologías*

- En cada iteración se parte de las historias priorizadas y se desarrollan (módulos de) ontologías
- Si algunas historias se solapan, se realiza un merge y se desarrolla una única ontología (o módulo)
- Se reusan patrones de diseño adecuados, que se especializan y/o componen (si es necesario)

### Fase 3: *Fase iterativa que integra incrementos (módulos) a la solución total*

- Ontologías o módulos se integran a la red de ontologías, identificando las relaciones de la nueva ontología con las demás (ej. mapping).

# SAMOD

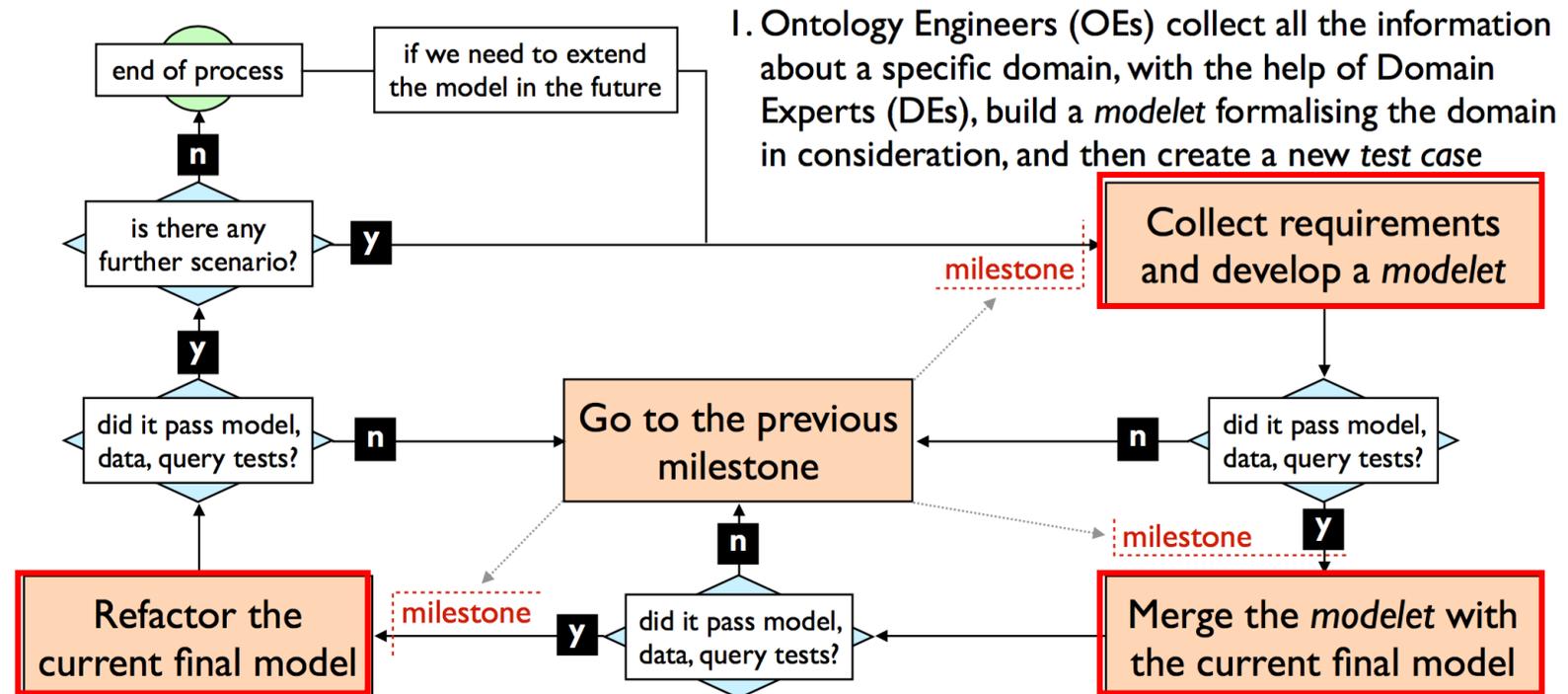
## Principios

En Paso 1:

- Mantener en lo posible pocos conceptos y relaciones
- Usar patrones de diseño
- Enfoque middle-out para modelar conceptos
- Mantener el modelo simple (sin agregar elementos innecesarios)
- Nombres de conceptos y relaciones autoexplicativos

En Paso 3:

- Reusar conocimiento
- Documentar (anotaciones, fuente/procedencia, descripciones en lenguaje natural)
- Enriquecer el modelo con inferencias



1. Ontology Engineers (OEs) collect all the information about a specific domain, with the help of Domain Experts (DEs), build a *modelet* formalising the domain in consideration, and then create a new *test case*

Go to the previous milestone

2. OEs merge the *modelet* of the new *test case* with the current model produced by the end of the last process iteration (first iteration: *modelet* becomes current model)

3. OEs refactor (important: reuse existing knowledge) the current model, focussing on the last part added in the previous step

Silvio Peroni. A simplified agile methodology for ontology development.

## Ciclo de vida ágil e iterativo

3 pasos que se repiten por cada escenario

# Ejemplo

**Propósito de la ontología** (uso esperado)

Asistir en el descubrimiento de recetas de diferentes sitios web.

## **Escenario (1)**

Se desea implementar una aplicación web de descubrimiento de recetas publicadas en diferentes sitios web. Para ello, se debe diseñar una ontología que asista a la aplicación en el descubrimiento de las recetas.

La ontología se instanciará con los datos de los sitios web de recetas. La aplicación de descubrimiento permitirá formular consultas, devolviendo (a través de la ontología) enlaces a recetas de diferentes sitios web como resultados.

La ontología debe ser fácilmente extendida con elementos externos adicionales, por ejemplo, información nutricional sobre ingredientes o información equipamiento de cocina.

## **Alcance**

*Historia de usuario 1 (HU1)*: obtener recetas por nombre, ingredientes y autor

*Historia de usuario 2 (HU2)*: obtener recetas por datos nutricionales (por ejemplo, cantidad de calorías)

(1) Pascal Hitzler, Adila Krisnadhi. A Tutorial on Modular Ontology Modeling with Ontology Design Patterns: The Cooking Recipes Ontology

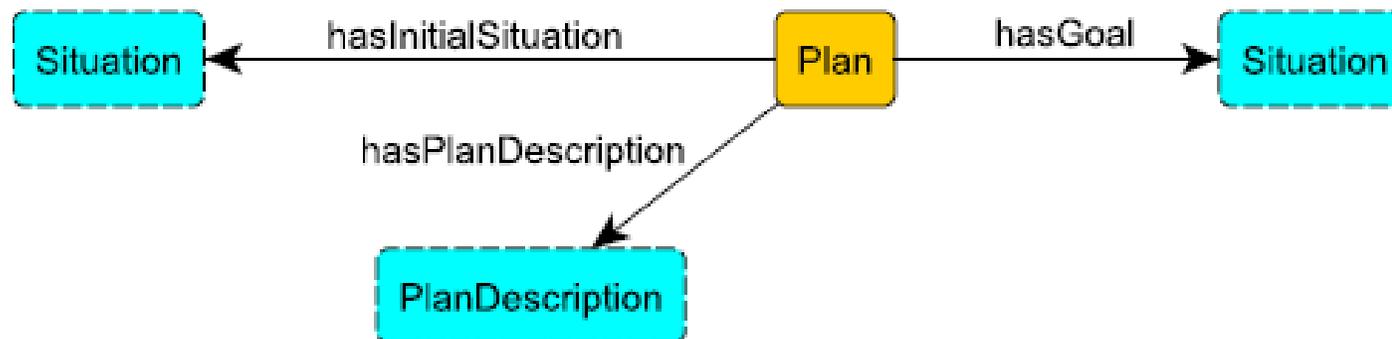
# Ejemplo – HU1

*Como* usuario de la aplicación de búsqueda de recetas,  
*quiero* poder consultar recetas por nombre, por una lista de ingredientes y por autor,  
*para* obtener las instrucciones de elaboración

## **Conceptualización**

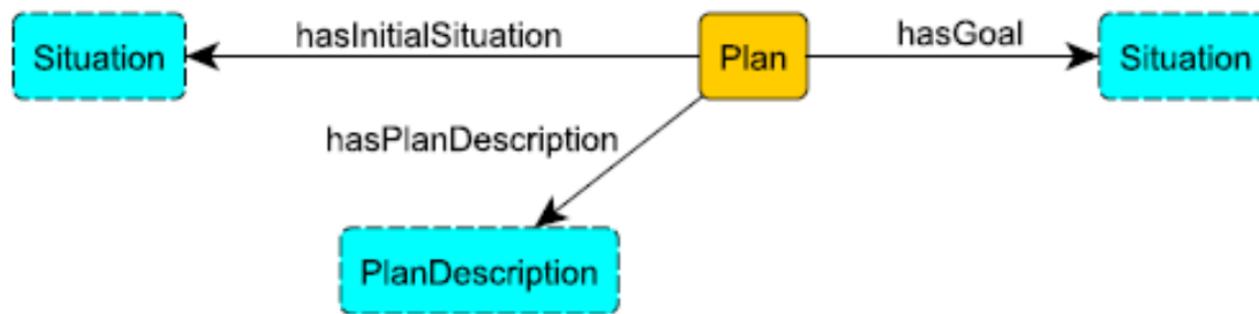
Principales términos: *Receta*, *Ingredientes*, *Instrucciones*, *Comida*

Patrón de diseño: *Basic plan ontology design Pattern*

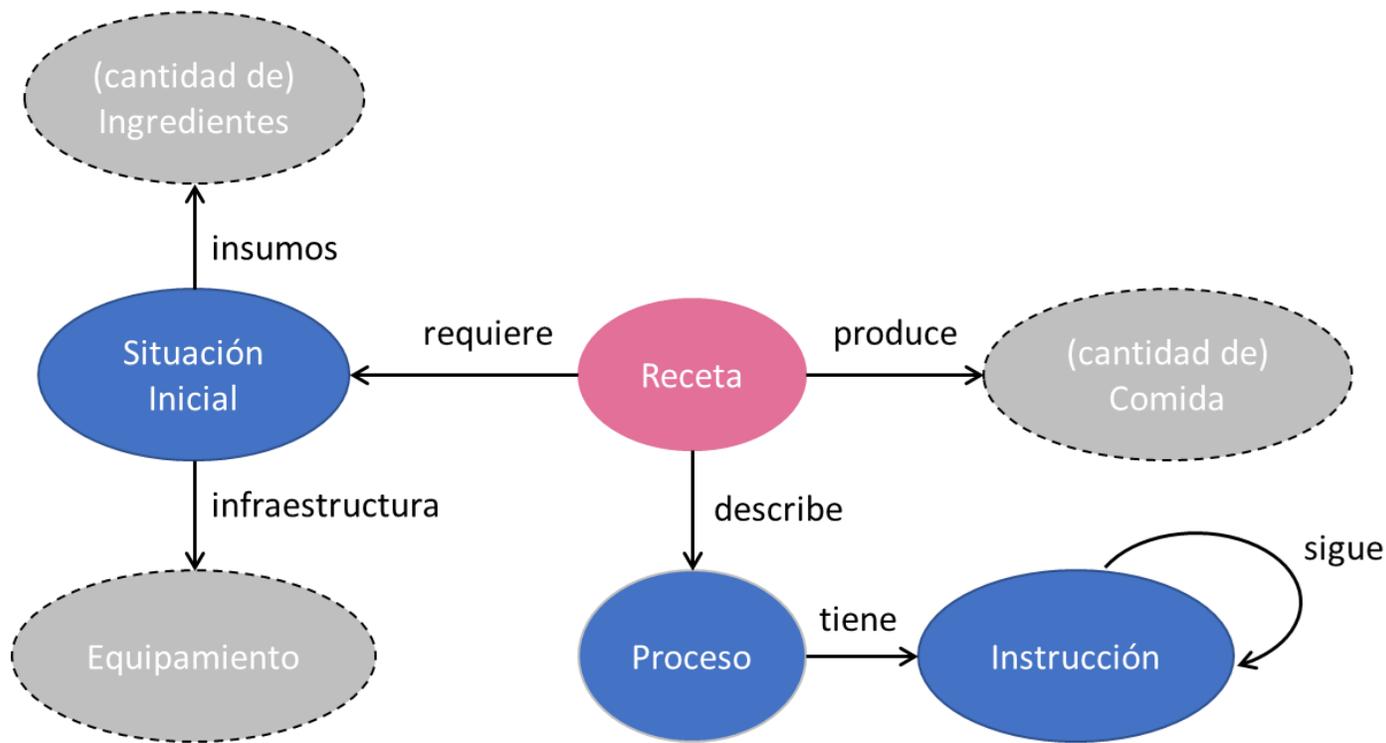
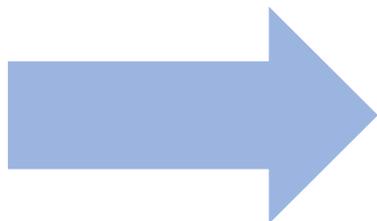


# Ejemplo – U1 - Conceptualización

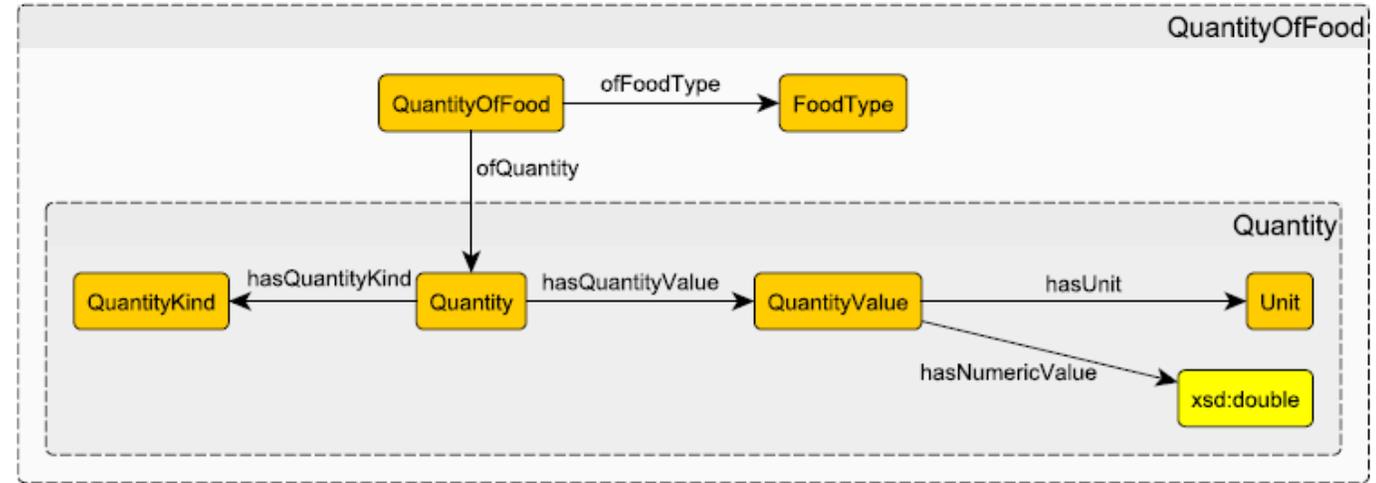
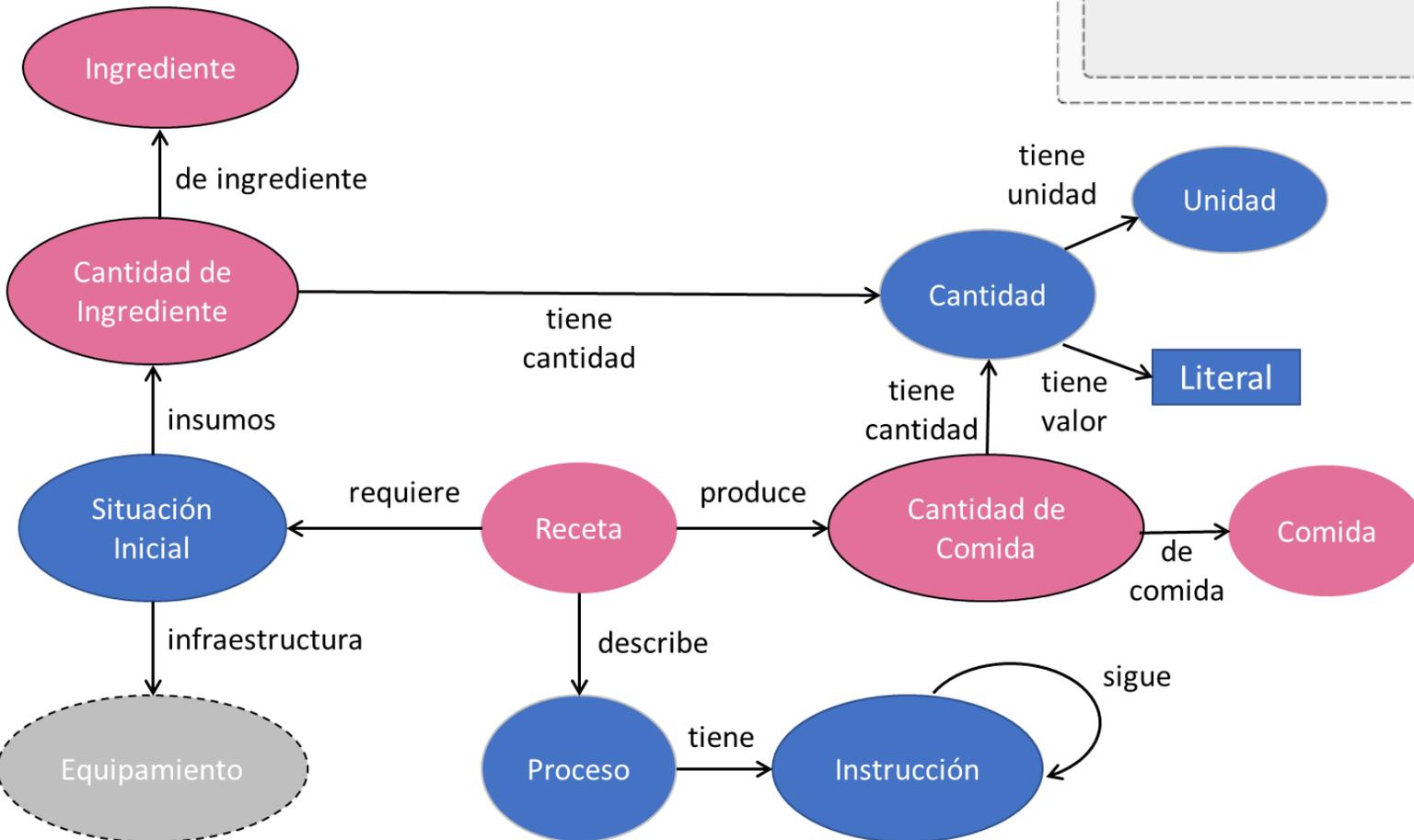
*Como* usuario del sitio de búsqueda de recetas *quiero* poder consultar recetas por nombre, por una lista de ingredientes y por autor, *para* obtener las instrucciones de elaboración



**Modelo basado en el patrón de diseño**



# Ejemplo – U1 - Conceptualización



# Ejemplo – HU1 – Formalización, implementación y testeo

$Receta \sqcap SituacionInicial \sqsubseteq \perp \quad Receta \sqcap Proceso \sqsubseteq \perp$

$Receta \sqcap Instruccion \sqsubseteq \perp \quad Proceso \sqcap Instruccion \sqsubseteq \perp$

$Receta \sqsubseteq \exists requiere.SituacionInicial$

$Receta \sqsubseteq =1 produce.CantidadDeComida \sqcap$

$\forall produce.CantidadDeComida$

$CantidadDeComida \sqsubseteq =1 deComida.CantidadDeComida \sqcap$

$\forall deComida.CantidadDeComida$

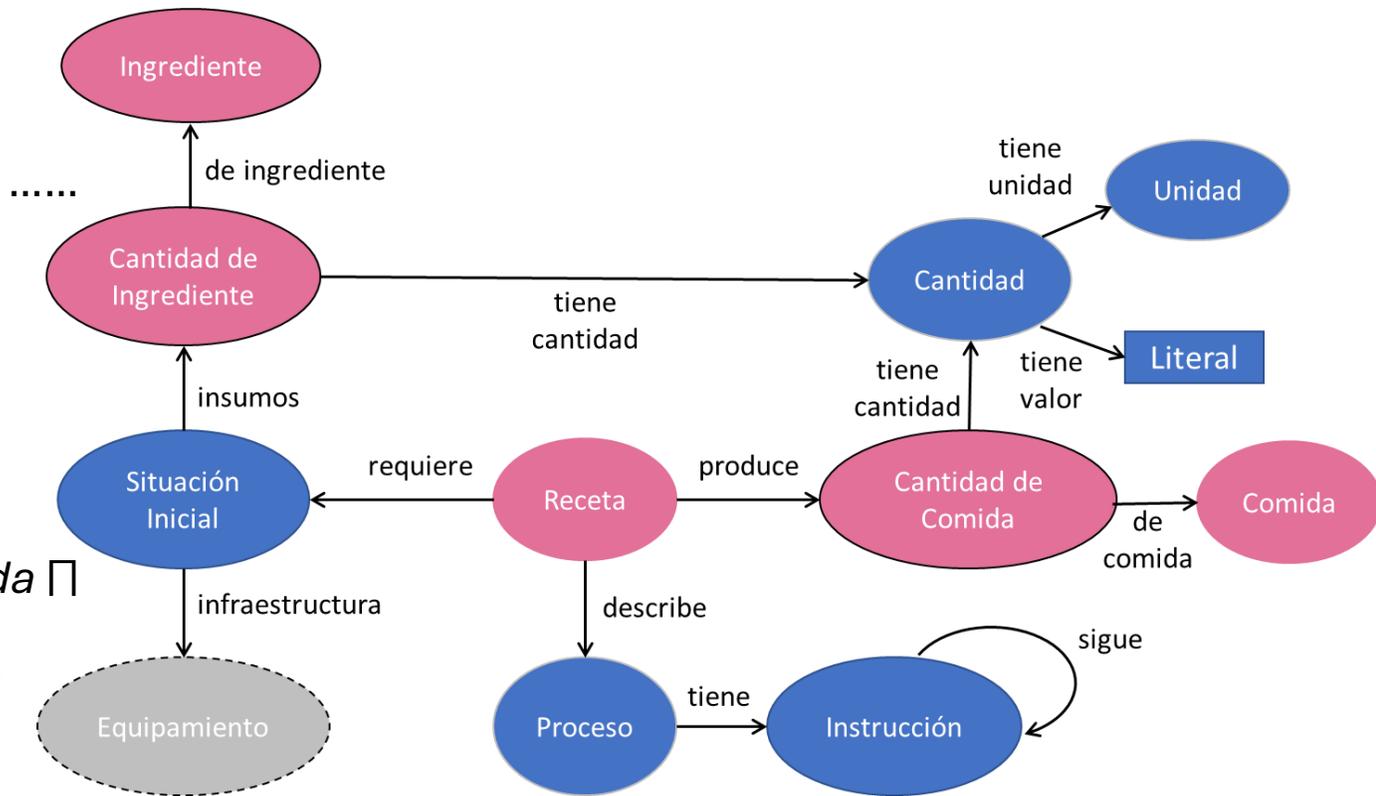
$Receta \sqsubseteq =1 describe.Proceso \sqcap \forall describe.Proceso$

$\exists describe.T \sqsubseteq Receta \quad T \sqsubseteq \forall describe.Proceso$

$Proceso \sqsubseteq =1 describe^{-}.Receta$

$Proceso \sqsubseteq \exists tiene.Instruccion \sqcap \forall tiene.Instruccion$

.....



# Ejemplo – HU1 – Formalización, implementación y testeo

$Receta \sqcap SituacionInicial \sqsubseteq \perp$     $Receta \sqcap Proceso \sqsubseteq \perp$

$Receta \sqcap Instruccion \sqsubseteq \perp$     $Proceso \sqcap Instruccion \sqsubseteq \perp$

$Receta \sqsubseteq \exists requiere.SituacionInicial$

$Receta \sqsubseteq =1 produce.CantidadDeComida \sqcap$

$\forall produce.CantidadDeComida$

$CantidadDeComida \sqsubseteq =1 deComida.CantidadDeComida \sqcap$

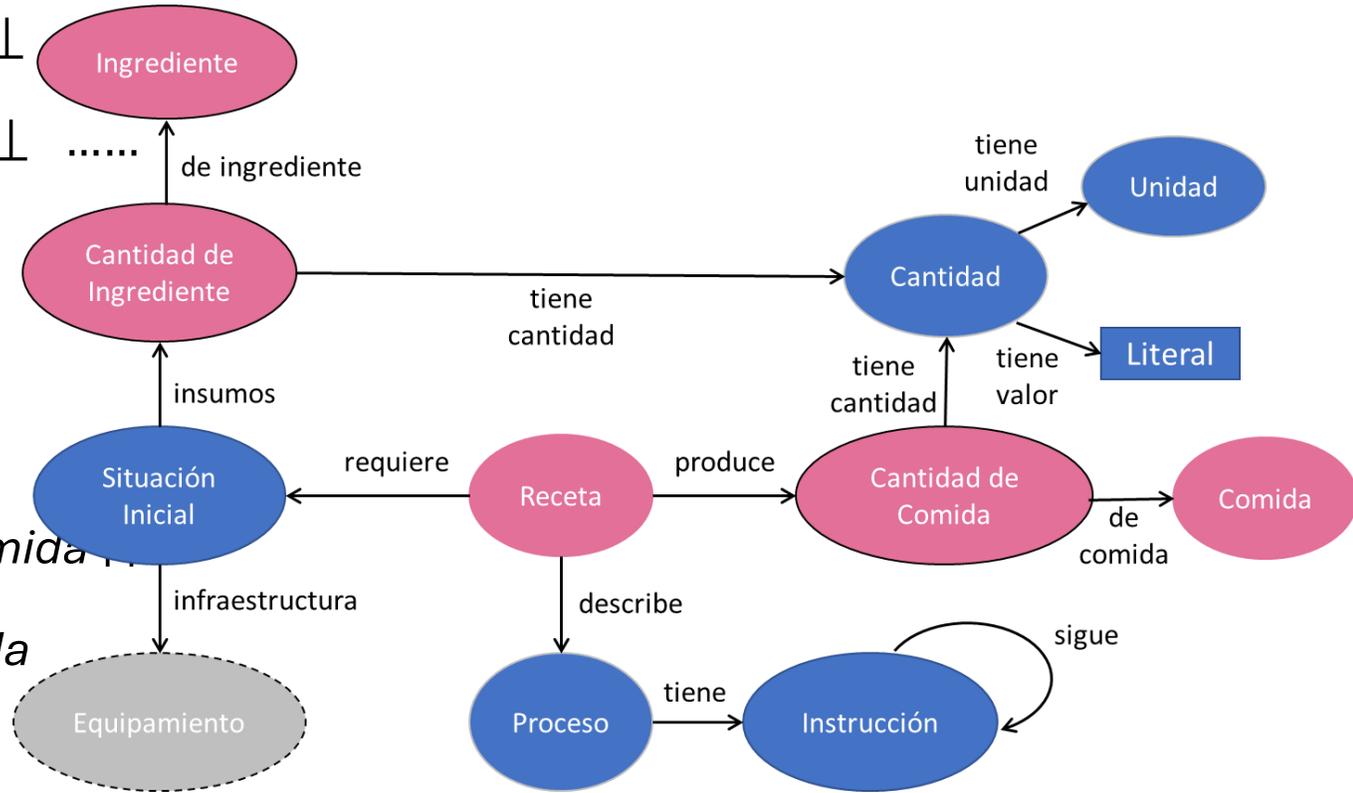
$\forall deComida.CantidadDeComida$

$Receta \sqsubseteq =1 describe.Proceso \sqcap \forall describe.Proceso$

$\exists describe.T \sqsubseteq Receta$     $T \sqsubseteq \forall describe.Proceso$

$Proceso \sqsubseteq =1 describe^- .Receta$

$Proceso \sqsubseteq \exists tiene.Instruccion \sqcap \forall tiene.Instruccion$



Implementar (archivo owl)

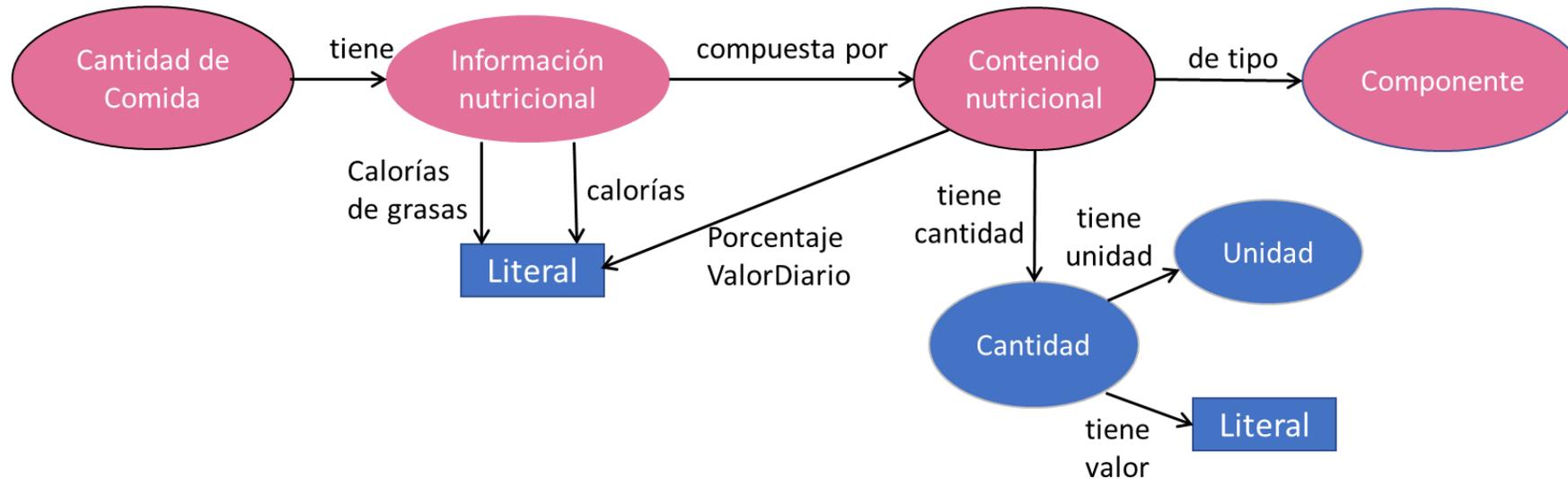
Ejecutar casos de prueba (consultas, inferencias)

# Ejemplo – HU2

Como usuario de la aplicación de búsqueda de recetas, quiero poder consultar recetas por datos nutricionales para obtener las instrucciones de elaboración

## Conceptualización

Principales términos: *Información nutricional*, *Contenido nutricional*, *calorías*, *porcentaje valor diario*

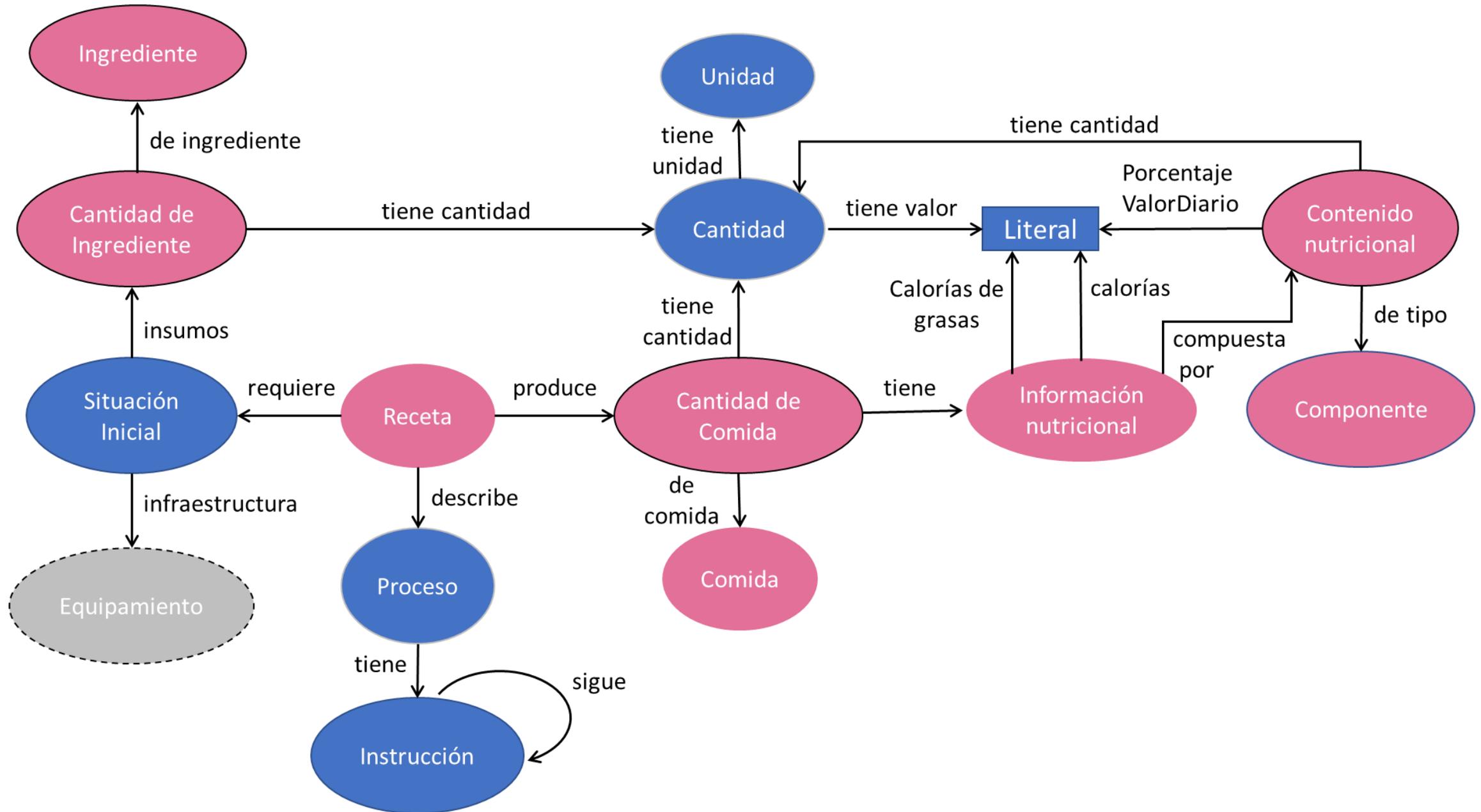


Información nutricional	
Tamaño de la porción 1/4 de taza (113 g)	
Porciones por envase 8	
Cantidad por porción	
Calorías 100	Calorías de las grasas 20
% de valor diario*	
<b>Grasa total</b> 2g	<b>3%</b>
Grasas saturadas 1.5g	<b>7%</b>
Grasas trans 0g	
<b>Colesterol</b> 10mg	<b>3%</b>
<b>Sodio</b> 460mg	<b>19%</b>
<b>Total de carbohidratos</b> 4g	<b>1%</b>
Fibra 0g	<b>0%</b>
Azúcares 4g	
<b>Proteína</b> 16g	
Vitamina A 0%	• Vitamina C 0%
Calcio 8%	• Hierro 0%

\* Los porcentajes de valores diarios se basan en una dieta de 2.000 calorías

Formalizar, implementar, ejecutar casos de prueba

# Integrando implementación HU2 a HU1



# Bibliografía

Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho. Ontological Engineering. Springer, 2003.

Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, Aldo Gangemi. Ontology Engineering in a Networked World. Springer, 2012.

Ontology Summit Communique.

[http://ontolog.cim3.net/wiki/OntologySummit2013\\_Communique.html](http://ontolog.cim3.net/wiki/OntologySummit2013_Communique.html)

E. Blomqvist, K. Hammar, and Valentina Presutti. Engineering ontologies with patterns - the extreme design methodology. IOS Press, 2016.

Silvio Peroni. A simplified agile methodology for ontology development. OWLED, 2016

Catálogo de patrones de diseño: <http://www.ontologydesignpatterns.org>

Valentina Presutti, Aldo Gangemi, Stefano David, Guadalupe Aguado de Cea, Mari Carmen Suárez-Figueroa, Elena Montiel-Ponsoda, María Poveda. D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies, 2005.