

# CommonKADS + SCRUM: Ingeniería del Conocimiento Ágil

Una combinación poderosa para desarrollar Bases de Conocimiento de Sistemas Expertos de forma rigurosa y flexible.

**Realizado por:**

**Khristian Flores: C.I.: 28.308.362**

**Duberth Farias: C.I.: 28.345.088**



# Ingeniería del Conocimiento y CommonKADS

La Ingeniería del Conocimiento (IC) es clave para los Sistemas Expertos (SE), enfocándose en adquirir, modelar, validar y gestionar conocimiento experto.

Utilizamos **CommonKADS**, una metodología estándar para sistemas basados en conocimiento, que modela el conocimiento de forma sistemática e independiente de la implementación técnica.

## Modelo de Conocimiento

Define conceptos, reglas, hechos y relaciones del dominio (ej.: "**SI** rizomas == "claramente diferentes" Y frondes == "Planos y en forma de lámina", **ENTONCES** Alga = "Caulerpa prolifera" (confianza=0.95).

## Modelo de Adquisición

Guía la extracción de conocimiento de expertos humanos (entrevistas, adquisición de documentos y claves dicotómicas).

## Modelo de Validación

Verifica la consistencia, completitud y precisión de la Base de Conocimiento (BC).

**Ventajas:** Estructura clara, reutilización de patrones y documentación robusta.



# Integración: CommonKADS y SCRUM

Combinamos CommonKADS con SCRUM para desarrollar la Base de Conocimiento (BC) de un Sistema Experto de forma rigurosa pero flexible. Esta integración aprovecha la solidez de CommonKADS en modelado de conocimiento y la adaptabilidad de SCRUM para gestionar incertidumbre y cambios.



## CommonKADS

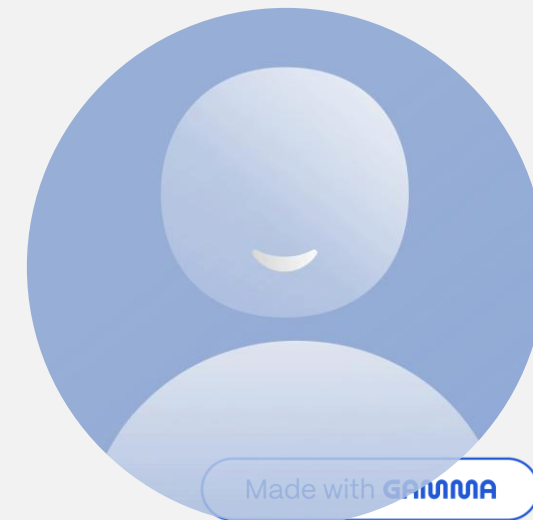
Gestiona el **contenido** (conocimiento experto): reglas, claves, lógica de inferencia.



## SCRUM

Gestiona el **proceso**: planificación, entregas iterativas, coordinación del equipo técnico.

CommonKADS se enfoca en **qué** debe saber el sistema, mientras que SCRUM se enfoca en **cómo** implementarlo.



# Roles y Responsabilidades Híbridas



## Knowledge Owner

Un experto del dominio (en nuestro caso, en Algas) valida reglas, aporta conocimiento experto, facilita claves dicotómicas, prioriza especies.



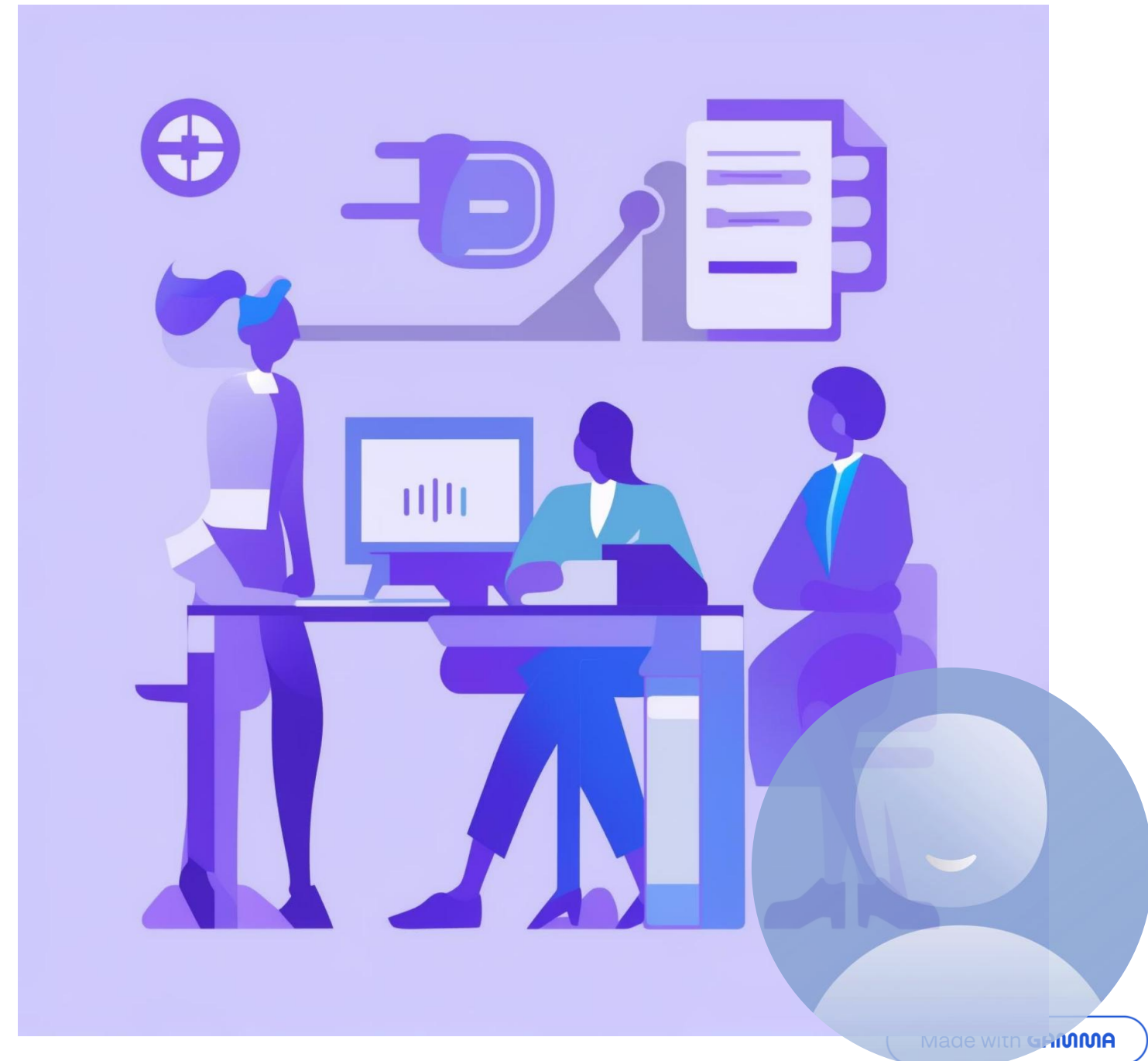
## SCRUM Master

Facilita sprints y la comunicación entre el Knowledge Owner y desarrolladores.



## Equipo de Desarrollo

Implementan motor de inferencia, base de conocimientos e interfaz en código.



# Fases de CommonKADS en Ciclos SCRUM

Modelo de Organización	Sprint 0: Definir alcance, objetivos y modelado inicial del Sistema Experto.
Modelo de Tareas	Sprint Planning: Priorizar tareas como "Crear módulo de motor de reglas en Python." en el backlog.
Modelo de Conocimiento	Sprints iterativos: Desarrollar fragmentos de la Base de Conocimiento.
Modelo de Adquisición	Reuniones diarias (daily): Validar reglas con expertos y ajustar basado en feedback.
Modelo de Validación	Sprint Review: Probar la Base de Conocimiento con casos reales y medir precisión.

# Planificación Ágil del Conocimiento

## Backlog como "Repositorio de Conocimiento"

Los "user stories" se convierten en **artefectos de conocimiento**.

**Ejemplo:** Story: Como usuario, quiero seleccionar características morfológicas".  
Tarea CommonKADS: crear una serie de preguntas con sus posibles opciones para sirvan para identificar una especie.

## Sprints Cortos (2-3 semanas)

Cada sprint entrega al menos 3 **especie de Caulerpa validada de la BC** (ej.: Caulerpa fastigiate, Caulerpa prolifera y Caulerpa verticillata).

**Criterio de Aceptación:** Exactitud >90% en casos de prueba.

# Retos y Soluciones

1

## Reto: Profundidad vs. Velocidad

CommonKADS requiere profundidad;  
SCRUM prioriza velocidad.

**Solución:** Usar **Sprint 0** para modelado inicial (30% del conocimiento), luego refinamiento iterativo.

2

## Reto: Validación Compleja

Validar conocimiento complejo en sprints cortos.

**Solución:** Incluir al experto del dominio en **dailies** rápidas (15 min) para resolver dudas críticas.

3

## Reto: Documentación vs. Agilidad

Equilibrar la documentación con la agilidad.

**Solución:** Generar **documentación mínima viable** (ej.: reglas en formato IF-THEN + ejemplos de uso).

# Beneficios Clave de la Combinación



## Flexibilidad

Incorporar nuevo conocimiento descubierto durante el proyecto (ej.: necesidad de agregar una nueva especie de *Caulerpa*).



## Calidad

Validación continua evita errores en la Base de Conocimiento (BC).



## Transparencia

El "product backlog" muestra el estado del conocimiento en tiempo real.

La combinación CommonKADS + SCRUM sigue este flujo:

**[Planificación KADS en Sprint 0] → [Sprints con: Adquisición → Modelado → Validación] → [Refinamiento continuo]**



# Conclusión: Calidad y Agilidad

La integración de CommonKADS y Scrum es una combinación poderosa para construir Sistemas Expertos de forma eficiente.

## CommonKADS

Estructura el "**qué**" (conocimiento experto riguroso).

## SCRUM

Gestiona el "**cómo**" (implementación iterativa y adaptativa).

