

1. DATOS INFORMATIVOS

Carrera: Ingeniería de Software

Asignatura: Análisis y Diseño de Software

Tema del taller: Pensamiento Crítico y Casos de Uso PlantUML

Docente: Jenny Ruiz

Integrantes: Kevin Asmal, Diego Delgado, Gabriel López, Marcelo Pareja

Fecha: 29/10/2025 Paralelo: 27835

2. DESARROLLO

Realizar con PlantUML un diagrama de casos de uso de los requerimientos del sistema Hotpoltademas de las preguntas de Pensamiento Crítico. El proceso se compondrán de 4 requerimientos importantes para el sistema, para entender cómo funciona y cómo se relaciona con herramientas de graficación cómo es PlantUml.

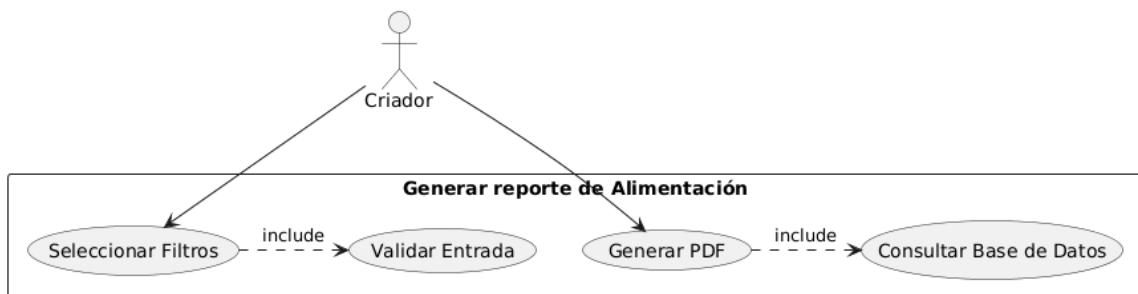
CU003.1 Generar Reporte de Alimentación

Autor: Gabriel López

Actor principal: Cuidador

Acciones Principales: Selección de Filtros, Generación de PDF

Diagrama UML



Código UML

@startuml

actor Criador

```
rectangle "Generar reporte de Alimentación"{
```

```
(Seleccionar Filtros) as (UC1)
```

```
(Generar PDF) as (UC2)
```

```
(Validar Entrada) AS (UC3)
```

```
(Consultar Base de Datos) as (UC4)
```

```
Criador --> (UC1)
```

```
Criador --> (UC2)
```

```
(UC1) .> (UC3) : include
```

```
(UC2) .> (UC4) : include
```

```
}
```

```
@enduml
```

Pensamiento Crítico

1. ¿Qué diferencias observas entre los casos de uso derivados de entrevistas o descripciones textuales y los casos de uso generados automáticamente en PlantUML?

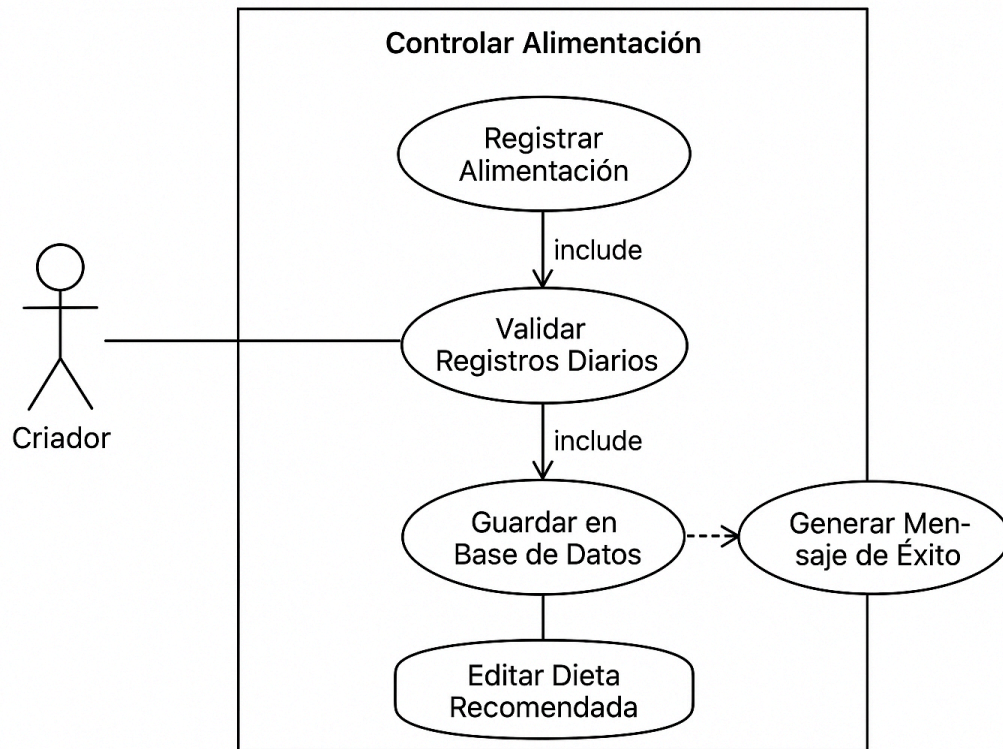
La primera diferencia y más grande es el tiempo que tarda generar un caso de uso y de refactorizar, los casos textuales pueden ser precisos en los contenidos que tienen pero pueden sentirse incompletos y difícil de comprender por como se observan, a diferencia del modelo visual que proporciona plantUML, a pesar de la dificultad de conocer como generar un diagrama con código, destaca en cómo permite representar correctamente el caso de uso planteado.

2. ¿De qué manera el uso de PlantUML facilita o limita el trabajo del analista al modelar los requisitos funcionales ?

PlantUML puede facilita en su mayoría el proceso de modelación, con solo comprender el funcionamiento básico y sus comandos, permite modelar con facilidad y rapidez, en caso de errores, es posible refactorizar el diagrama para encapsular acciones nuevas o eliminadas del

modelo, pero puede limitar por el formato en el que genera los diagramas, y para poder generar diagramas con mayor complejidad, sería necesario usar funciones más avanzadas de plantUML y los formatos disponibles para exportación.

CU001.7 Controlar alimentación.



@startuml

actor Criador

rectangle "Controlar Alimentación" {

(Registrar Alimentación) as (UC1)

(Validar Registros Diarios) as (UC2)

(Editar Dieta Recomendada) as (UC3)

(Guardar en Base de Datos) as (UC4)

(Generar Mensaje de Éxito) as (UC5)

Criador --> (UC1)

Criador --> (UC3)

(UC1) .> (UC2) : include

(UC1) .> (UC4) : include

(UC4) .> (UC5) : include

}

@enduml

Pensamiento Crítico

1. ¿Qué diferencias observas entre los casos de uso derivados de entrevistas o descripciones textuales y los casos de uso generados automáticamente en PlantUML?

Los casos de uso textuales permiten un análisis detallado y documentan cada paso, pero pueden resultar extensos y difíciles de visualizar. En cambio, PlantUML representa gráficamente las relaciones y dependencias entre acciones, facilitando la comprensión global del proceso de “Controlar Alimentación” de forma rápida y clara.

2. ¿De qué manera el uso de PlantUML facilita o limita el trabajo del analista al modelar los requisitos funcionales?

PlantUML facilita la modelación porque permite crear diagramas completos con pocas líneas de código y mantener una trazabilidad visual de los casos de uso. Sin embargo, puede limitar cuando se requieren diagramas más complejos o personalizaciones visuales avanzadas, ya que exige conocer su sintaxis y comandos específicos.

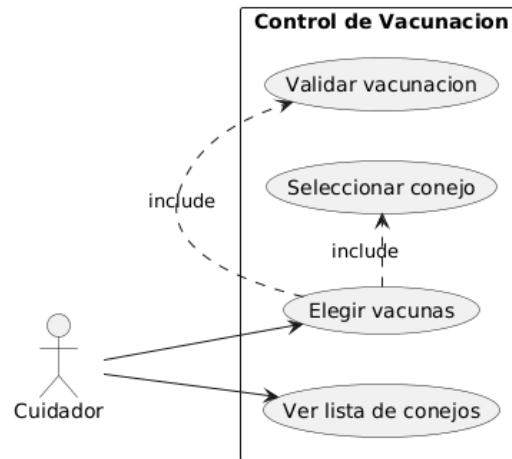
CU 001.6 Controlar la vacunación

Autor: Marcelo Pareja

Actor principal: Criador

Actividades principales: Ver listado de conejos, aplicar vacunas, validar vacunación.

Diagrama de PlantUML



Código PlantUML

@startuml

left to right direction

actor "Cuidador" as cuidador

rectangle "Control de Vacunacion"{

usecase "Ver lista de conejos" as UC1

usecase "Elegir vacunas" as UC2

usecase "Validar vacunacion" as UC3

usecase "Seleccionar conejo" as UC4

}

cuidador --> UC1

cuidador --> UC2

UC2 .->UC3: include

UC2 .->UC4: include

@enduml

Pensamiento Crítico

1. ¿Qué diferencias observas entre los casos de uso derivados de entrevistas o descripciones textuales y los casos de uso generados automáticamente en PlantUML?

Los casos de uso derivados de entrevistas y/o descripciones textuales, tienen una mayor libertad para personalizarlos facilitando el entendimiento del cliente, sin embargo, toman mucho tiempo en elaborar y se debe cuidar la organización de los elementos. Los casos de uso en PlantUML no tienen un nivel de personalización tan alto, pero se organizan automáticamente, y crean el caso de uso en cuestión de segundos.

2. ¿De qué manera el uso de PlantUML facilita o limita el trabajo del analista al modelar los requisitos funcionales?

PlantUML facilita en gran medida el trabajo del analista al modelar requisitos funcionales, ya que le permite trasladar sus ideas al modelo rápidamente a través del código. Si el analista desea refinar el modelo, simplemente cambia el código y el modelo se actualiza al instante.

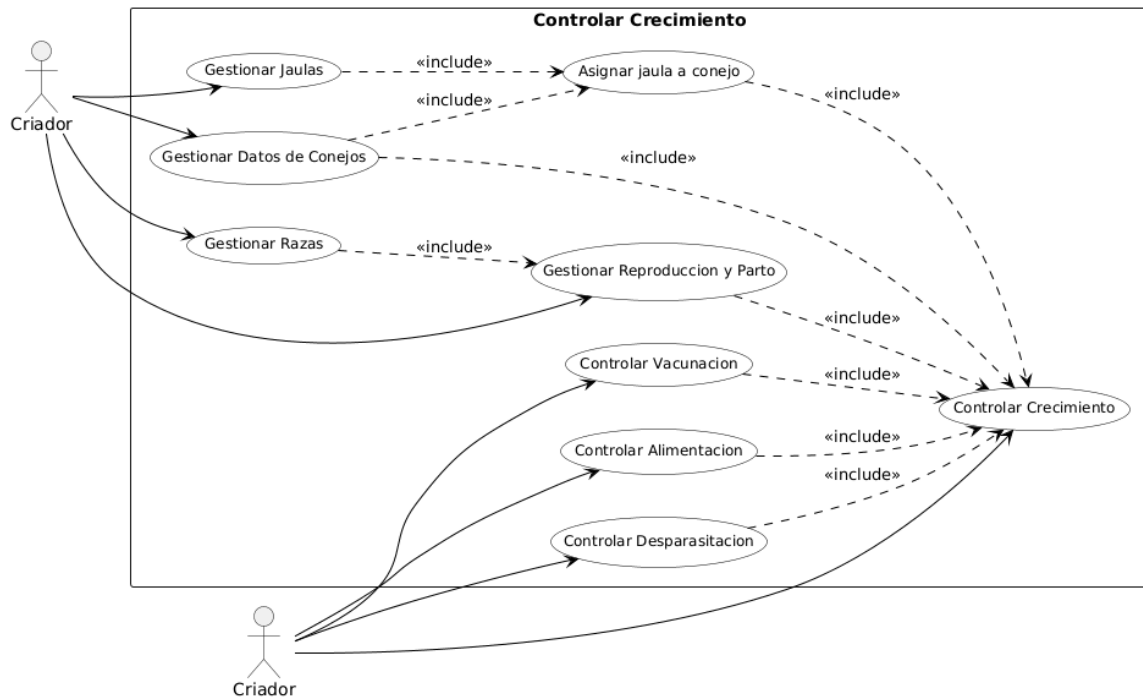
CU 001.8 Controlar Crecimiento

Autor: Diego Delgado

Actor principal: Criador

Actividades principales: Calcular la edad del conejo, detallar incremento o decremento en el peso de cada conejo.

Diagrama de PlantUML



@startuml

' Dirección principal

left to right direction

skinparam shadowing false

skinparam packageStyle rectangle

skinparam usecase {

 BackgroundColor White

 BorderColor Black

 ArrowColor Black

 FontSize 12

}

skinparam actorStyle stickman

'--- Actores (misma etiqueta visible, alias distintos) ---

actor "Criador" as C1

actor "Criador" as C2

rectangle "Controlar Crecimiento" {

 usecase "Gestionar Jaulas" as UC1

 usecase "Asignar jaula a conejo" as UC2

 usecase "Gestionar Datos de Conejos" as UC3

 usecase "Gestionar Razas" as UC4

 usecase "Gestionar Reproduccion y Parto" as UC5

 usecase "Controlar Vacunacion" as UC6

 usecase "Controlar Alimentacion" as UC7

 usecase "Controlar Desparasitacion" as UC8

 usecase "Controlar Crecimiento" as UC9

}

'--- Conexiones del actor izquierdo ---

C1 --> UC1

C1 --> UC3

C1 --> UC4

C1 --> UC5

'--- Conexiones del actor derecho ---

C2 --> UC6

C2 --> UC7

C2 --> UC8

C2 --> UC9

'--- Relaciones <<include>> (uso flechas punteadas estándar "..>") ---

UC1 ..> UC2 : <<include>>

UC3 ..> UC2 : <<include>>

UC4 ..> UC5 : <<include>>

UC7 ..> UC9 : <<include>>

UC8 ..> UC9 : <<include>>

UC6 ..> UC9 : <<include>>

UC3 ..> UC9 : <<include>>

UC5 ..> UC9 : <<include>>

UC2 ..> UC9 : <<include>>

'--- Ayudas de posicionamiento (vínculos ocultos para forzar actor derecho) ---

' Estos enlaces ocultos ayudan a que C1 quede a la izquierda y C2 a la derecha

C1 -[hidden]-> UC1

UC9 -[hidden]-> C2

@enduml

Pensamiento Crítico

3. ¿Qué diferencias observas entre los casos de uso derivados de entrevistas o descripciones textuales y los casos de uso generados automáticamente en PlantUML?

Los casos de uso obtenidos de entrevistas o descripciones textuales suelen ser más detallados y reflejan mejor las necesidades reales del usuario, ya que provienen de su experiencia directa. En cambio, los casos de uso generados en PlantUML son más estructurados y visuales, pero pueden perder parte del contexto o de los matices que surgen en la comunicación con el usuario.

4. ¿De qué manera el uso de PlantUML facilita o limita el trabajo del analista al modelar los requisitos funcionales ?

PlantUML facilita el trabajo del analista porque permite crear diagramas de forma rápida y sencilla usando código, lo que ayuda a documentar los requisitos funcionales de manera clara y fácil de actualizar. Sin embargo, puede ser un poco limitado para quienes prefieren trabajar de forma visual, ya que requiere conocer su sintaxis para ajustar los diagramas.

3. CONCLUSIONES

- El uso de herramientas de modelado son necesarias en los procesos de desarrollo de software, facilita la generación de modelos gráficos con código simple, plantUML es un formato gratuito que permite un diseño estructurado pero fácil de aprender y generar. Con ello será posible facilitar en el caso de diagramas de caso de uso la implementación de una idea, diseñar un proceso formal para el sistema y facilita la distribución de el entendimiento de los procesos a computarizar.
- El uso de PlantUML en el caso **CU001.7 – Controlar Alimentación** permite visualizar de forma clara las acciones del criador y las validaciones automáticas del sistema. La herramienta facilita la comprensión de procesos como el registro, validación de raciones y control de la dieta, haciendo más eficiente la comunicación entre analistas y desarrolladores.
- El uso de PlantUML en el caso **CU001.6 - Controlar Vacunación** permite generar un modelo que represente de manera precisa las acciones y actores involucrados. Además permite realizar cambios y observar la evolución del modelo sobre la marcha.

- PlantUML es una herramienta práctica que facilita la creación y presentación de modelos funcionales, permitiendo al analista enfocarse en la lógica del sistema más que en el diseño visual. Su uso promueve claridad y estandarización, aunque su valor depende del análisis previo, ya que solo traduce las ideas del analista.

4. RECOMENDACIONES

Se recomienda seguir utilizando PlantUML para representar otros casos de uso del sistema, mantener consistencia en los diagramas y agregar los módulos complementarios (reportes y control de crecimiento) para mejorar la trazabilidad y la integración visual del sistema completo.

Combinar el uso de PlantUML con técnicas de levantamiento de información como entrevistas o análisis de documentos, para asegurar que los diagramas reflejan fielmente las necesidades del usuario y mantengan una comunicación efectiva entre analistas y desarrolladores.

5. REFERENCIAS

Escriba las fuentes utilizadas según el formato APA 7.0, por ejemplo:

Autor, A. A. (Año). *Título del libro o artículo*. Editorial o fuente. URL