

# Curso de Ingeniería de Software

## Unidad 4

# Administración de proyectos

Guadalupe Ibargüengoitia

Hanna Oktaba

# Entradas a esta unidad

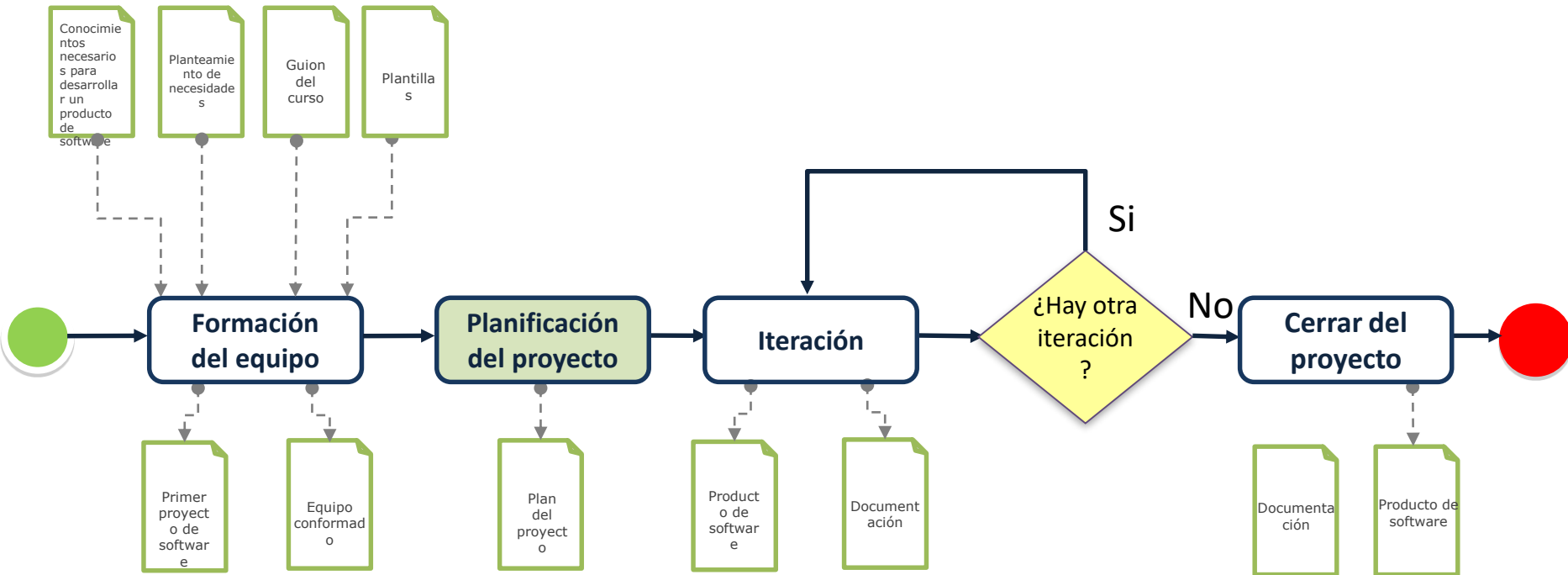
- **Condiciones**

- Equipo conformado
  - Con canales y formas de comunicación síncrona definidos.
  - Con repositorio común de documentos creado

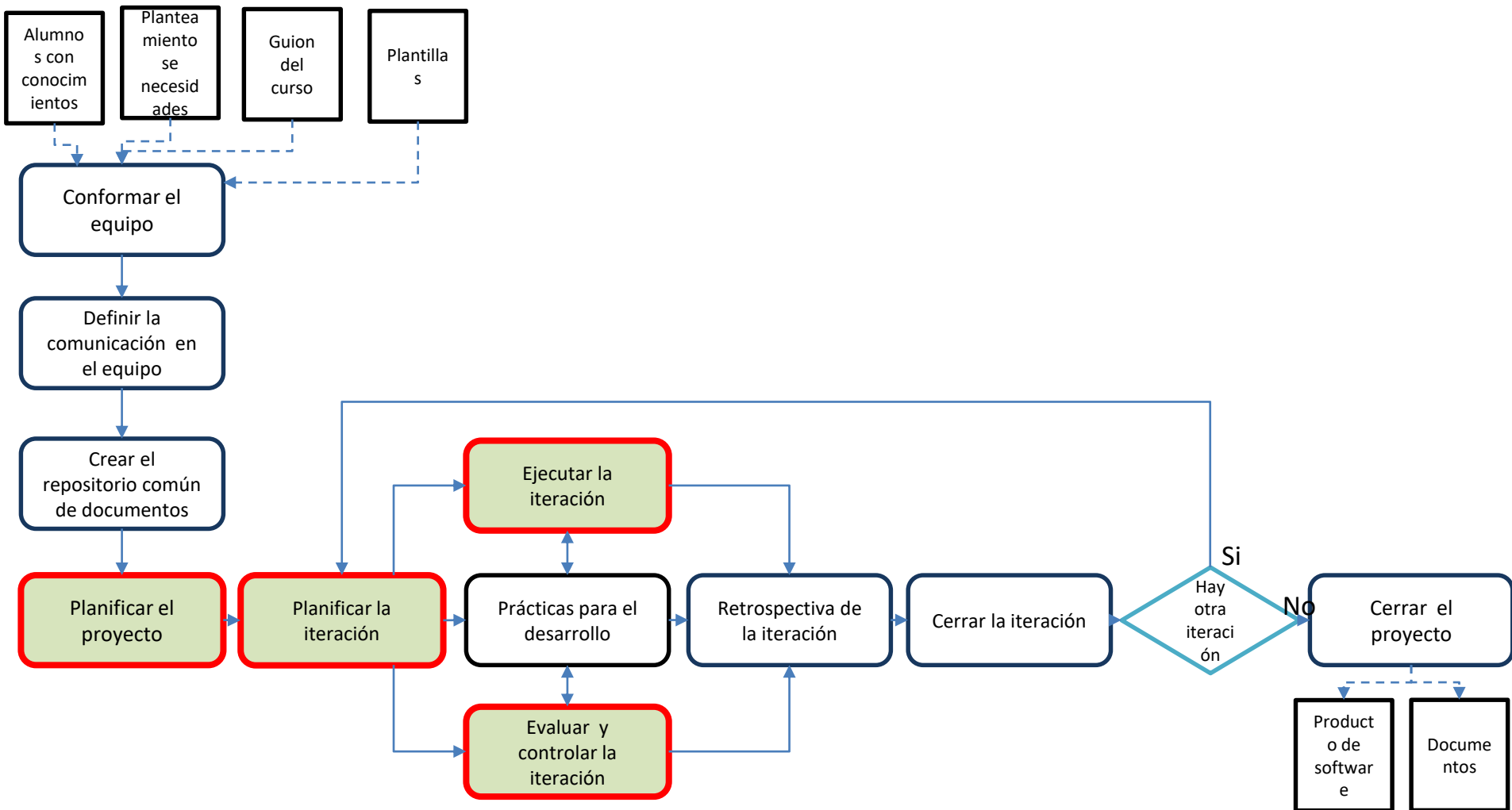
- **Productos de trabajo**

- *Primer proyecto de software*, con los datos del equipo, proyecto, forma de comunicación y repositorio
  - *Planteamiento de necesidades*
  - *Guion del curso*
- *Plantilla del Plan del proyecto.*

# Aplicación del método inicial de desarrollo de software (MIDS)



# Prácticas administrativas

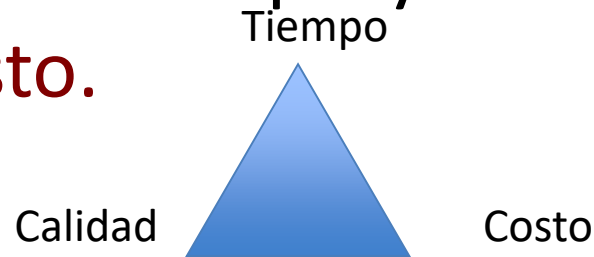


# Concepto de Proyecto

- Un *proyecto* se puede definir como el esfuerzo de una organización para desarrollar, introducir o evaluar un resultado único.
- Es un esfuerzo que se lleva a cabo durante un periodo de tiempo delimitado, con un equipo de trabajo asignado y con recursos disponibles para producirlo. (PMBOK)

# Administración de Proyecto

- La “*Administración de Proyectos*” es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos. (PMBOK, 2013)
- Requisitos principales de un proyecto son: **calidad, tiempo y costo.**



# Administración de proyecto de software

- Tiene por objetivo establecer y llevar a cabo sistemáticamente un conjunto de actividades a fin de lograr el objetivo del proyecto de software en calidad, tiempo y costo.

(ISO/IEC29110)

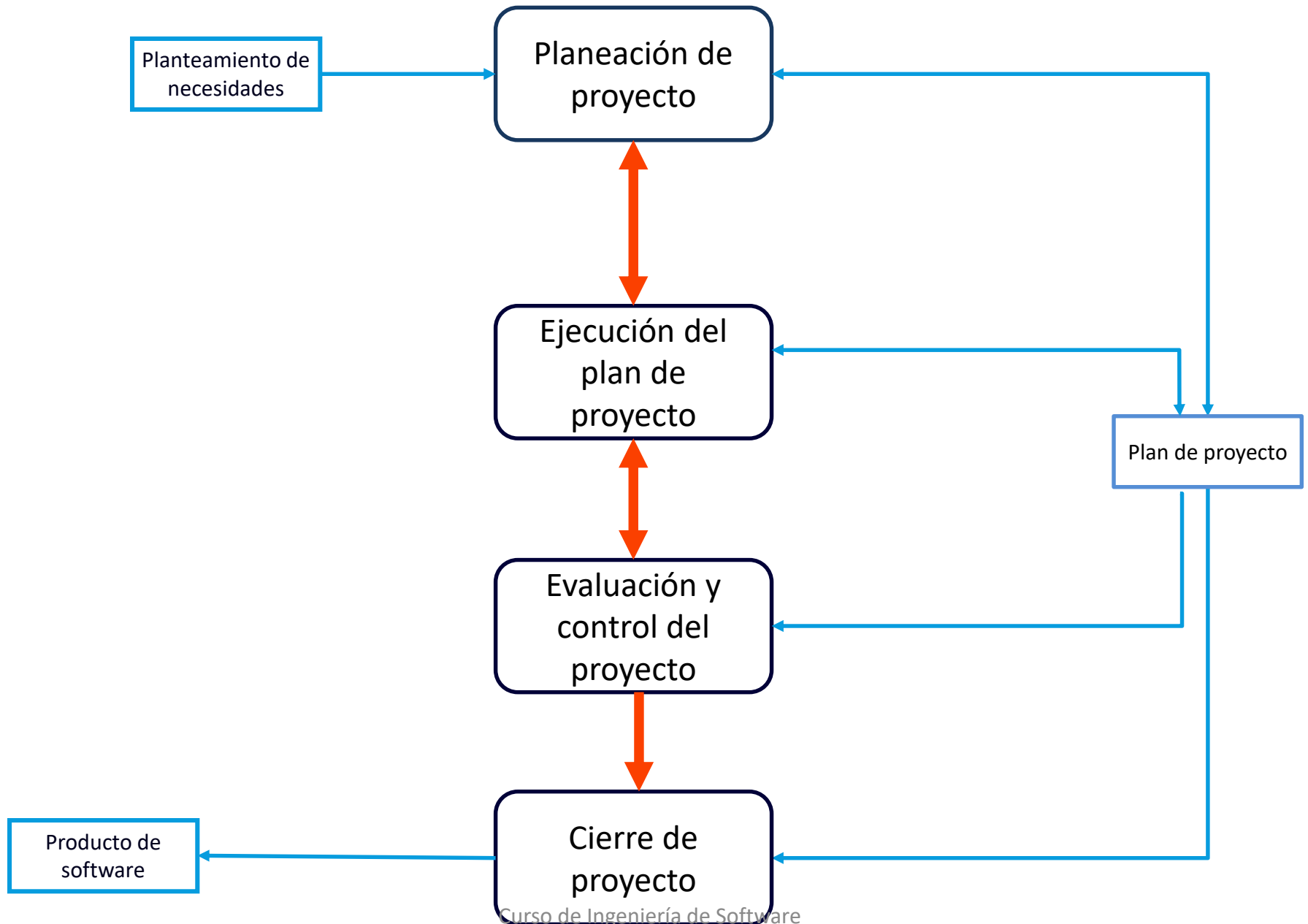
# Administración de proyecto

## ISO/IEC 29110 Perfil básico

- Entrada: *Planteamiento de Necesidades*
- Salida: *Plan del proyecto*
- Actividades principales son:
  - Planificar el proyecto
  - Ejecutar el Plan del proyecto
  - Evaluar y controlar el proyecto
  - Cerrar el proyecto



# Actividades del proceso de administración de proyecto



# Planificar el proyecto

- La planificación del proyecto se inicia con el entendimiento de sus objetivos y restricciones.
- Se definen qué actividades se deben llevar a cabo en un proyecto, quién las debe hacer y cuándo.
- El resultado de la actividad de planificar el proyecto es el documento llamado *Plan del proyecto*.

# Plan del proyecto

- Los **objetivos del proyecto** acordes con las necesidades del cliente.
- Las **fechas de inicio y fin** del proyecto.
- La **lista de los entregables**, que son los documentos que serán entregados al cliente, junto con el código del producto de software.
- La **fecha y forma de cómo se entregará** el producto de software acordadas con el cliente.
- La **identificación de las funcionalidades** que tendrá el sistema o producto de software.
- **Restricciones no funcionales** (por ejemplo: uso de cierto lenguaje de programación o base de datos, entre otros).

# Diagrama general de casos de uso

- Una de las técnicas para identificar las funcionalidades de un producto de software, se llama *casos de uso* (Jacobson I., 2012)
- Es una vista gráfica más general y abstracta del sistema de software.
- Representa las funcionalidades generales del software.
- Define qué tipo de usuarios utilizarán el software.
- El diagrama es un conjunto de casos de uso.

# Diagrama general de casos de uso

- Esta técnica se utiliza para comprender las necesidades o los requerimientos funcionales y a partir de ellos se diseña, implementa y prueba el software.
- La importancia del *Diagrama general de casos de uso* reside en que las prácticas de desarrollo se pueden guiar por los casos de uso (Rosenberg Doug, 2001)

# Componentes de un Caso de uso

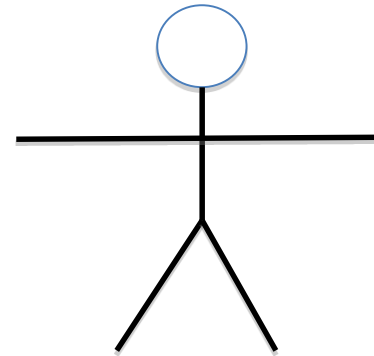
- Caso de uso

- Funcionalidad expresada iniciando con verbo



- Actor

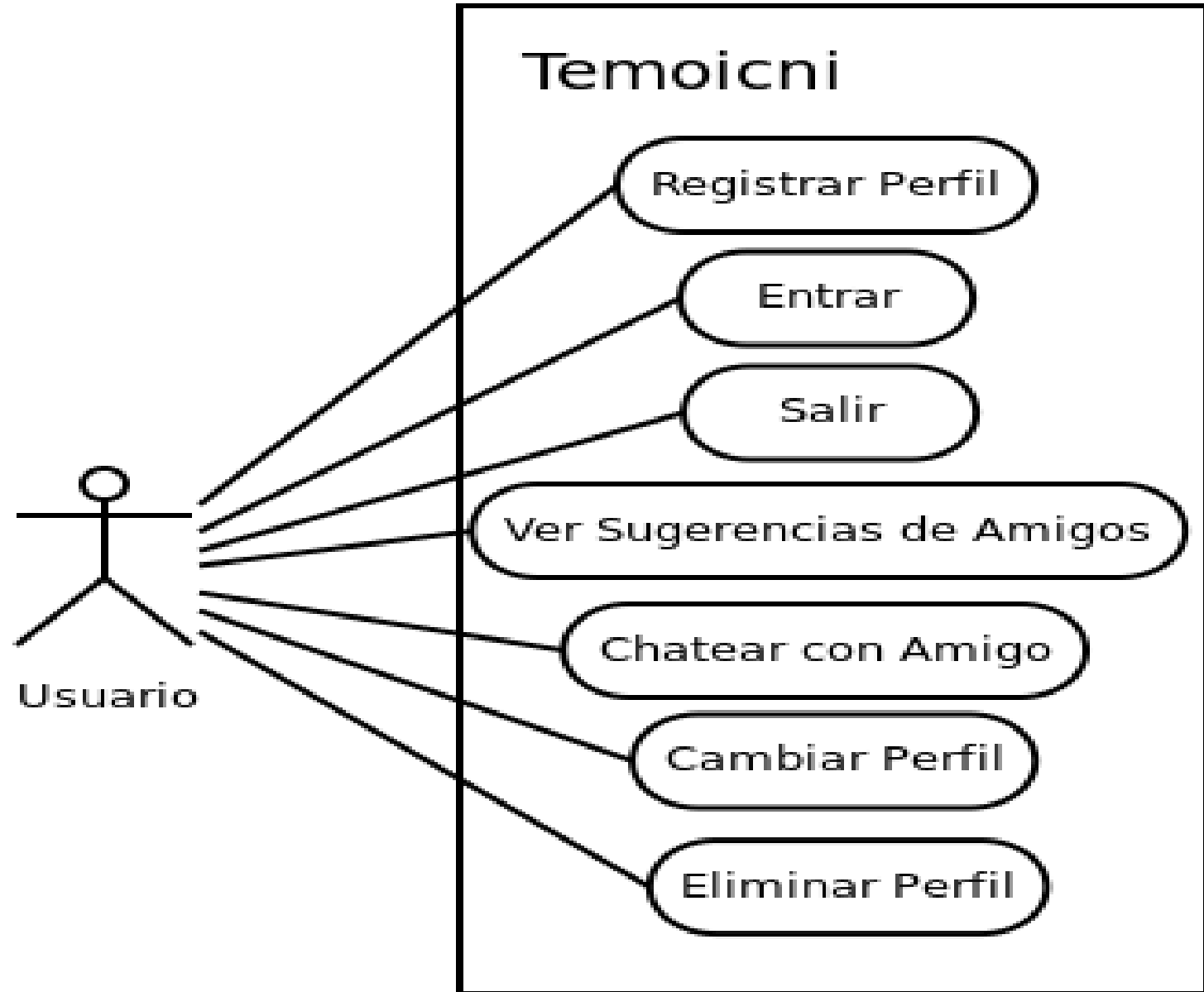
- Nombre de usuario en singular



- Relación de asociación

Usuario

# Ejemplo del Diagrama general de casos de uso



# Práctica Administrativa 1

## Planificar el proyecto

PA1	Práctica
<b>Planificar el proyecto</b>	
<b>Objetivos</b>	
<ul style="list-style-type: none"> <li>• Generar el Plan del proyecto.</li> </ul>	
<b>Entrada</b>	<b>Resultado</b>
<b>Condiciones</b> <ul style="list-style-type: none"> <li>• Equipo de trabajo conformado</li> </ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"> <li>• Primer proyecto de software               <ul style="list-style-type: none"> <li>○ Planteamiento de necesidades</li> <li>○ Guion del curso</li> </ul> </li> </ul>	<b>Condiciones</b> <ul style="list-style-type: none"> <li>• Proyecto planificado</li> </ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"> <li>• Plan del proyecto               <ul style="list-style-type: none"> <li>○ Nombre del proyecto</li> <li>○ Objetivo del proyecto</li> <li>○ Fecha de inicio y fin</li> <li>○ Entregables</li> <li>○ Fecha y forma de entrega del product de software</li> <li>○ Identificación de las funcionalidades del producto de software.</li> </ul> </li> </ul>



# Práctica Administrativa 1

## Planificar el proyecto

Actividades
<ol style="list-style-type: none"><li>1. Identificar las funcionalidades del producto.</li><li>2. Hacer el Plan del proyecto.</li></ol>
Herramientas
<ul style="list-style-type: none"><li>• Herramienta de diagramación en UML</li></ul>

# Elementos del Plan del Proyecto

- Los puntos que incluye el Plan del Primer Proyecto de Software:
  - Objetivo del proyecto: **basado en** Planteamiento de Necesidades.
  - Fechas: **las fecha de inicio y fin del proyecto** definidas en el Guión del Curso.
  - Entregables: descripción **de lo que se entregará al final del proyecto**, estos son los documentos resultantes de cada práctica del desarrollo y el código de software. (ver nombres de las plantillas y el Guión del Curso)

# Elementos del Plan del Proyecto

- Fechas y forma de entregar el producto: están definidas en el Guión del Curso.
- Diagrama general de casos de uso

Llenar con estos datos la **Plantilla del Plan del Primer Proyecto de Software**

# TA1. Técnica para Identificar las funcionalidades del producto

- Uno de los elementos para hacer el *Plan del proyecto* es la identificación de las funcionalidades del producto de software.
- Esta tarea la vamos a hacer construyendo el *Diagrama general de casos de uso*

# TA1. Técnica para Identificar las funcionalidades del producto

- **Pasos para identificar a los actores y los casos de uso:**
  - Identificar los actores del sistema. Para identificar a un actor se revisa el *Planteamiento de necesidades* a fin de identificar qué tipo de usuarios interactuarán con el software.
  - Identificar los casos de uso para cada actor. Para cada actor que ya identificaste, buscas en el *Planteamiento de necesidades*, qué es lo que podrá hacer ese actor con el sistema.

# TA1. Técnica para Identificar las funcionalidades del producto

- Dibujar el Diagrama general de casos de uso
  - Para cada actor identificado, dibujar el actor con la herramienta.
  - Para cada caso de uso identificado, dibujar un óvalo en el diagrama y asignarle nombre que inicia con un verbo en infinitivo, tal como fue identificado en el *Planteamiento del problema*.
  - Para cada actor dibujar una línea recta entre el actor y los casos de uso que puede ejecutar. Un actor puede ejecutar muchos casos de uso y un caso de uso puede ser efectuado por muchos actores.
  - Todos los casos de uso identificados se enmarcan en un rectángulo para delimitar el alcance del sistema.

# Planificar la iteración

- Los métodos ágiles de desarrollo de software (ver Unidad 2) proponen que el desarrollo sea *iterativo*.
- El desarrollo se divide en varias *iteraciones* de tiempos cortos.
- En cada iteración se realizan las actividades que permiten **entregar un producto con funcionalidades parciales**, que el cliente puede probar.
- El producto va **incrementando sus funcionalidades** con las entregas en cada iteración, hasta satisfacer completamente las necesidades del cliente.

# Definición del alcance de cada iteración

- Se escogen qué funcionalidades se implementarán en esa iteración.
- Criterios de selección de casos de uso para una iteración
  - Seleccionar los que son prioritarios para el cliente
  - Seleccionar los que no dependen de los que no se seleccionaron
  - Seleccionar por lo menos un caso de uso por integrante del equipo
- La duración de cada iteración depende del contexto de cada proyecto y las prioridades del cliente.



# Tablero

- Para ayudar al equipo a tener **visibilidad** de lo que hay que hacer en cada iteración y permitir que se le dé **seguimiento**, usaremos un *Tablero*.
- Este *Tablero* es una versión simplificada de una **técnica ágil** llamada **Kanban** (Anderson, 2010).
- El objetivo del *Tablero* es permitir la **visibilidad del avance** en las actividades de desarrollo de todos los miembros del equipo y poder saber **quién** está trabajando en cada parte del proyecto.

# Tablero digital Trello

- Es un tablero digital en línea que el equipo puede usar y consultar

[trello.com](https://trello.com)

- Se crea el *Tablero* para el proyecto e iteración:
  - El encabezado el nombre del proyecto,
  - el número de la iteración
  - las fechas de inicio y fin de la iteración.

- El *Tablero* consta de tres columnas con los encabezados:
  - *Por hacer* (To do en inglés)
  - *Haciéndose* (Doing)
  - *Hecho* (Done)
- En la columna de *Por hacer* se van colocando tarjetas

# Tarjeta de trabajo

- Una tarjeta tiene:
  - **Título.** Debe ser un nombre representativo del trabajo.
  - **Descripción de las actividades a hacer.**
  - **Responsable de realizar la actividad.**
  - **Fecha de entrega,** para cuando se espera que el trabajo esté terminado.

# Tarjeta de trabajo- ejemplo

## Dibujar el Diagrama general de casos de uso

- Dibujar los actores
- Dibujar los casos de uso
- Establecer las relaciones entre actores y casos de uso
- Enmarcar el alcance de los casos de uso

Responsable: Luis

Fecha de entrega: 24 de marzo

# *Método Inicial de Desarrollo de Software*

- Práctica Administrativa PA2

Planificación de la iteración

# Práctica Administrativa 2

## Planificar la iteración

PA1	Práctica
Planificar la iteración	
Objetivos	
<ul style="list-style-type: none"><li>• Planificar el alcance y preparar la iteración.</li></ul>	
Entrada	Resultado
<b>Condiciones</b> <ul style="list-style-type: none"><li>• Se iniciará una iteración</li></ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• Primer proyecto de software<ul style="list-style-type: none"><li>○ Repositorio compartido</li></ul></li><li>• Plan del proyecto</li></ul>	<b>Condiciones</b>  <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• Tablero de la iteración creado para el alcance de iteración</li><li>• Repositorio compartido incluyendo una carpeta para la iteración</li></ul>

# Práctica Administrativa 2

## Planificar la iteración

### Actividades

1. Construir el Tablero de la iteración (TA3)
2. Definir el alcance de la iteración (TA4)
3. Crear la carpeta de la iteración en el repositorio común para el resguardo de documentos de la iteración (TA5)

### Herramientas

- Herramienta para el Tablero
- Herramienta para repositorio de documentos



## TA3. Técnica para construir el *Tablero* de la iteración.

- Para crear el *Tablero* para cada iteración existen herramientas en web que ayudan a automatizarlo.
- Se recomienda hacer una búsqueda de este tipo de herramientas para crear tableros Kanban que sean gratuitos y que se podrán configurar según las necesidades del curso.
- Algunas herramientas para construir tableros son: [kanbanflow.com](https://kanbanflow.com), [leankit.com](https://leankit.com), [trello.com](https://trello.com), [asana.com](https://asana.com)

# Construcción del Tablero

- Los pasos para crear el *Tablero* son:
  - Si es la primera iteración, el **Responsable de la colaboración**, en conjunto con el equipo **selecciona la herramienta** para construir el *Tablero* y asegura que **todos se capacitan** en su uso.
  - El **Responsable de la colaboración** **construye** el *Tablero* para el equipo.
  - El **Responsable de la colaboración** pone en el *Tablero*, en la columna ***Por hacer una tarjeta***

### Crear Tablero de la iteración I

- Poner el encabezado
- Dar acceso al equipo
- Dar acceso al docente
- Responsable: Nombre (Responsable de la colaboración)
- Fecha de entrega: fecha de inicio de la iteración

- Cuando inicia la construcción del *Tablero*, pasa la tarjeta a la columna *Haciéndose*
- Una vez que haya realizado todos los pasos, pasa la tarjeta a la columna *Hecho*

# Iteraciones en el Plan del Proyecto

- El primer proyecto se va a desarrollar **en dos iteraciones**:
  - **Primera iteración**, se construirá el producto de software para una selección de **funcionalidades prioritarias**.
  - **Segunda iteración**, se incluirán funcionalidades restantes y se refinarán las funcionalidades ya implementadas.
  - **Para definir el alcance de cada iteración** se debe generar el **diagrama general de casos de uso** (ver la Técnica A2. Diagrama general de casos de uso) aplicando los criterios de selección.

## TA4. Técnica para definir el alcance de la iteración.

- Al inicio de cada iteración se define su alcance como un subconjunto de casos de uso del *Diagrama general de casos de uso*, por lo que se hace un diagrama específico para esa iteración.

- Para establecer cuáles casos de uso se incluirán en la iteración se deben aplicar los criterios:
  - Casos de uso que son prioritarios para el cliente
  - Casos de uso independientes de otros
  - Casos de uso que dependen de los que ya están implementados
  - Cada miembro del equipo sea responsable de por lo menos un caso de uso.

- El **Responsable técnico** convoca a una reunión del equipo siguiendo la técnica TS6 con el objetivo de escoger los casos de uso para el alcance de la iteración ejecutando la técnica TA4.
- Cada **Desarrollador** coloca en el *Tablero*, en la columna de *Por hacer*, una *tarjeta azul* por cada caso de uso en el alcance de la iteración del cual es responsable con fecha de terminación, el fin de la iteración.

## TA5. Técnica para crear la carpeta de la iteración en el repositorio común para el resguardo de documentos de la iteración

- Al inicio de cada iteración se coloca una carpeta nueva en el repositorio compartido, con el identificador de la iteración.
- Esta carpeta servirá para resguardar todos los documentos generados durante la iteración.
- Si se trata de la primera iteración, se colocan en esa carpeta, las plantillas para todos los entregables.



- El **Responsable de la calidad** es el encargado de personalizar y asegurarse que estén correctos los productos.
- El **Responsable de la colaboración** se encarga de llevar el control de las versiones de los documentos y que estén resguardados en el repositorio en la carpeta correspondiente.

- El **Responsable de la colaboración** coloca la tarjeta de *Crear la carpeta de la iteración* en el *Tablero* en la columna **Por hacer**, moviendo esa tarjeta según avanza en la realización de esa actividad.
- El **Responsable de la calidad** coloca la tarjeta *Iniciarizar los documentos en la carpeta de la nueva iteración* en la columna **Por hacer** del *Tablero*, moviendo esa tarjeta según avanza en la realización de esa actividad.

# Ejecutar el Plan del proyecto

- Dentro de la Administración de proyectos, la actividad de *Ejecutar el proyecto* tiene como objetivo asegurarse que se lleve a cabo el plan en tiempo y forma.
- Hay que darle seguimiento a las actividades que están realizando los miembros del equipo.

# Ejecutar el Plan del proyecto

- Para **saber** si el plan se cumple, hay que dar el **seguimiento** a su ejecución:
  - Evidencia de las reuniones del equipo de trabajo (ejemplo: reuniones diarias)
  - Registro del estado del progreso actualizado (tablero de seguimiento)
  - Repositorio del proyecto actualizado y respaldado
  - Solicitudes cambios que se analizan y evalúan revisando los impactos a los costos, calendario y requerimientos técnicos que necesitan. Si se aprueban, **se documentan**.

- Cuando se está trabajando de forma iterativa, en la ejecución se tiene el **subconjunto de las actividades** a realizar en el proyecto, restringido al **alcance de la iteración**

# *Método Inicial de Desarrollo de Software*

- Práctica Administrativa PA3

Ejecutar el Plan de la iteración

## Práctica Administrativa 3

### Ejecutar el plan de la iteración

PA3	Práctica
Ejecutar el plan de la iteración	
<b>Objetivos</b>	
<ul style="list-style-type: none"><li>• Asegurarse que el <i>Plan del proyecto</i> se lleve a cabo en tiempo y forma según lo planeado para la iteración.</li></ul>	
<b>Entrada</b>	<b>Resultado</b>
<b>Condiciones</b> <ul style="list-style-type: none"><li>• Iteración en ejecución</li></ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• <i>Tablero</i> de la iteración</li></ul>	<b>Condiciones</b> <ul style="list-style-type: none"><li>• Iteración en ejecución</li></ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• <i>Tablero</i> actualizado</li></ul>

# Práctica Administrativa 3

## Ejecutar el plan de la iteración

Actividades
1. Ejecutar lo planeado para la iteración revisando y actualizando el <i>Tablero</i> (TA6)
Herramientas
● Herramienta para el <i>Tablero</i>



- En la **reunión diaria del equipo**, en 15 min, todos contestan 3 preguntas y actualizan el estado del tablero de seguimiento según sus avances.
- Las preguntas son:
  - ¿Qué hice desde la última reunión?
  - ¿Qué haré?
  - ¿Qué dificultades encontré?
- Con base en las respuestas, el **responsable de la colaboración actualiza el tablero** moviendo las tarjeta a *Hecho* o generando nuevas

## TA6. Técnica para ejecutar lo planeado para la iteración revisando y actualizando el *Tablero*

- En las reuniones diarias del equipo, cada miembro **muestra su avance** reflejado en el **estado de sus tarjetas** en las columnas del *Tablero*.
- Según su **rol**, se agregan nuevas tarjetas al *Tablero* de acuerdo con sus **responsabilidades**
- El **Responsable de la colaboración** se asegura que **todos actualicen** el *Tablero* moviendo sus tarjetas según el estado del trabajo de cada día.

- De esta manera **todos sabrán** en qué están **trabajando los demás**, podrán establecer **comunicación** con quien tenga trabajo **relacionado**, o del que **dependa** su actividad.
- Todos pueden **ver el avance de sus compañeros**, notar si hay **retrasos** y tratar de **resolverlos**.
- Esta actividad se ejecutará diariamente a partir de la Unidad 5.

# Evaluar y controlar el proyecto

- El objetivo de la actividad de Administración de proyectos llamada **Evaluar y Controlar el proyecto**, es asegurarse que las actividades que se propusieron en el plan se lleven a cabo, que **se cumplan con los compromisos** establecidos y se tomen **las acciones pertinentes** ante **situaciones especiales**

# *Método Inicial de Desarrollo de Software*

- Práctica Administrativa PA4

Evaluar y controlar la iteración

Práctica Administrativa 4  
Evaluar y controlar la iteración

PA1	Práctica	
Evaluar y controlar el proyecto		
Objetivos		
<ul style="list-style-type: none"><li>• Evaluar el avance de la iteración en el <i>Tablero</i> y controlar las dificultades encontradas.</li></ul>		
Entrada	Resultado	
<b>Condiciones</b> <ul style="list-style-type: none"><li>• Iteración en ejecución</li></ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• <i>Tablero</i> de la iteración</li></ul>	<b>Condiciones</b> <ul style="list-style-type: none"><li>• Iteración en ejecución</li></ul> <b>Productos de trabajo</b> <ul style="list-style-type: none"><li>• <i>Tablero</i> de la iteración actualizado</li></ul>	

Curso de Ingeniería de Software

# Práctica Administrativa 4

## Evaluar y controlar la iteración

### Actividades

1. Evaluar el estado del progreso de la iteración a través del *Tablero*, identificando dificultades y retrasos. Proponer soluciones para controlarlas y actualizar el *Tablero* con las actividades necesarias (TA7)

### Herramientas

Herramientas para el *Tablero*

## TA7. Técnica para evaluar y controlar el estado del progreso de la iteración a través del *Tablero*

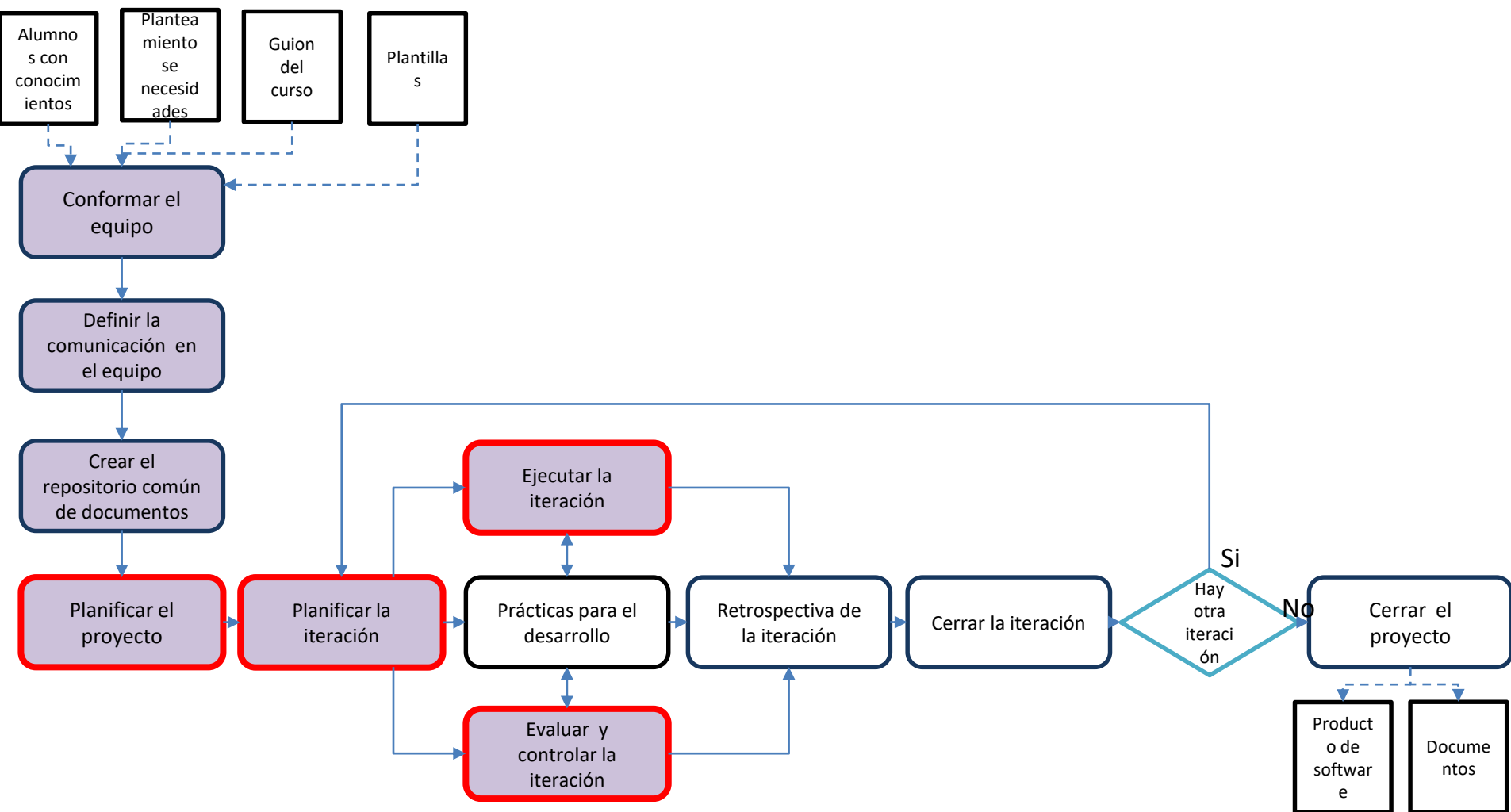
- Si al revisar el progreso de la iteración en el *Tablero* se detectan retrasos o dificultades, entre todos los miembros del equipo se analizan las razones y, de común acuerdo, se proponen actividades correctivas para remediarlas.
- Los retrasos y dificultades encontradas pueden ocasionar la creación de nuevas tarjetas con actividades para rectificarlos



- Si las dificultades son de origen técnico, el **Responsable técnico** busca la manera para apoyar a los que están atrasados, incorporando nuevas tarjetas en el *Tablero* con las actividades necesarias.
- Si el **Responsable de la calidad** revisando los productos generados por el equipo detecta defectos, crea una tarjeta de corrección en el *Tablero* y la asigna al responsable del defecto.
- Esta actividad de aprendizaje se ejecutará diariamente a partir de la Unidad 5.

# Cierre de la iteración y del proyecto

- Las actividades administrativas de cierre de la iteración y del proyecto se presentarán en la Unidad 10, cuando cerremos la iteración y el proyecto.



# Resultados de esta unidad

- **Condiciones**
  - Proyecto planificado
- **Productos de trabajo**
- *Plan del proyecto*
  - Nombre del proyecto
  - Objetivo del proyecto
  - [Fechas](#) de inicio y fin
  - [Entregables](#)
  - [Fecha y forma de entrega final del producto](#) de software
  - Identificación de las funcionalidades del producto de software
- Repositorio compartido con la carpeta para la iteración
- *Tablero* de la iteración

# Referencias

- Ambler Scott W. (2005). *The elements of UML 2.0 Style*. Cambridge Univertity Press.
- Anderson, D. J. (2010). *Kanban. Successful evolutionary change for your technology business*. Blue Hole Press.
- Brooks F. P. (1995). *The Mytical man-month and other essays on Software Engineering*. Addison-Wesley 2a ed.
- Dumke, R. A. (2011). *Cosmic Function Points: Theory and Advances Practices*. CRC Press.
- ISO/IEC29110. (2011). *29110-5-1-2Software Engineering-lifecycle Profiles for Very Small Entities Management and Engineering Guidde. s.1. Software Engineering*.
- Jacobson I. (2012). *Uses Case 2.0*.
- Jacobson I., C. M. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach*. Addison-Wesley.
- Jalote P. (2002). *Software Project Management in Practice*. Addison Wesley.
- PMBOK. (2013). *Project Mnagement Body of Kowledged*. PMI.
- Pressman R:S. (n.d.). *Ingeniería de software. Un enfoque práctico*. Mc Graw Hill.
- Rosenberg Doug, S. K. (2001). *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*. Addison Wesley.
- Rumbaugh J., I. J. (1998). *The Unified Modeling Language. Reference Manual*. Addison Wesley.
- Schmuller J. (s.f.). *Aprendiendo UML en 24 horas*. Prentice Hall.