Desarrollo de Software I

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación

Mauricio Gaona
mauricio.gaona@correounivalle.edu.co
Profesor

2023-I

Agenda

01 **RESUMEN**
Aspectos generales vistos en la clase anterior.

02 **REQUERIMIENTOS**
Requerimientos para el desarrollo de una aplicación de software.

03 **ACTIVIDADES PROXIMA CLASE**

Objetivos del curso

Objetivo general

Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo de aplicaciones de software.

- Metodología
- Evaluación
- Contenido
- Herramientas a utilizar en el curso

Herramientas

Algunas herramientas requeridas

Ambiente de trabajo: AV o MV

Herramientas de desarrollo : **Front end**
HTML5
JavaScript y CSS3

Back-end

Python
Django

Servidor Web: Apache, nginx

Bases de datos: PostgreSQL, Oracle

Herramientas de Gestión ([Google Docs, Jira, Taiga])
Manejador de versiones : GitHub , Bitbucket

IDE

https://azure.microsoft.com/es-es/free/students/

https://aws.amazon.com/es/education/awseducate/

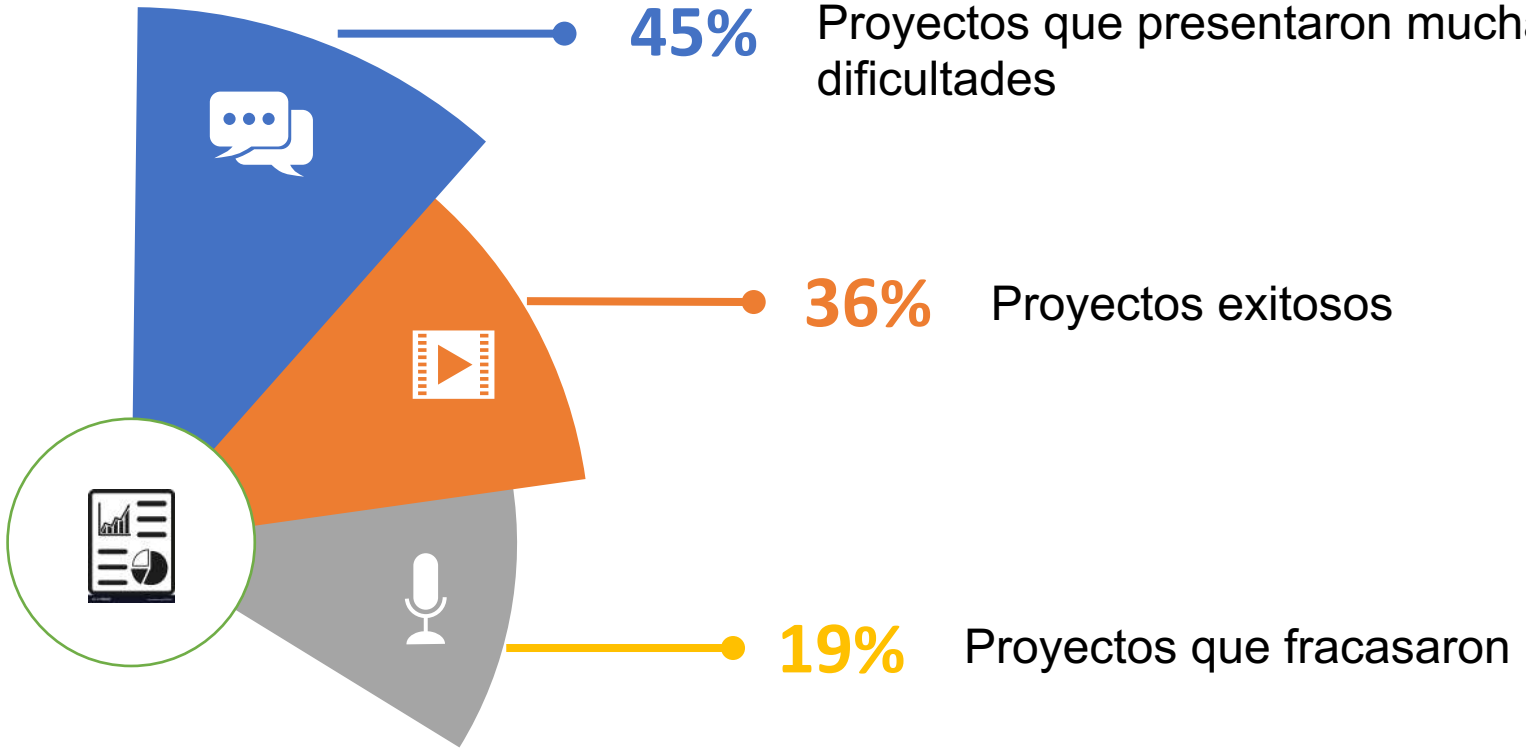
https://education.github.com/

SonarLint https://marketplace.visualstudio.com

Estado actual de la industria del software

Conceptos: **Crisis del software**

Reporte (2020) Caos Report Standish Group (encuesta a nivel mundial a empresas que hacen software)

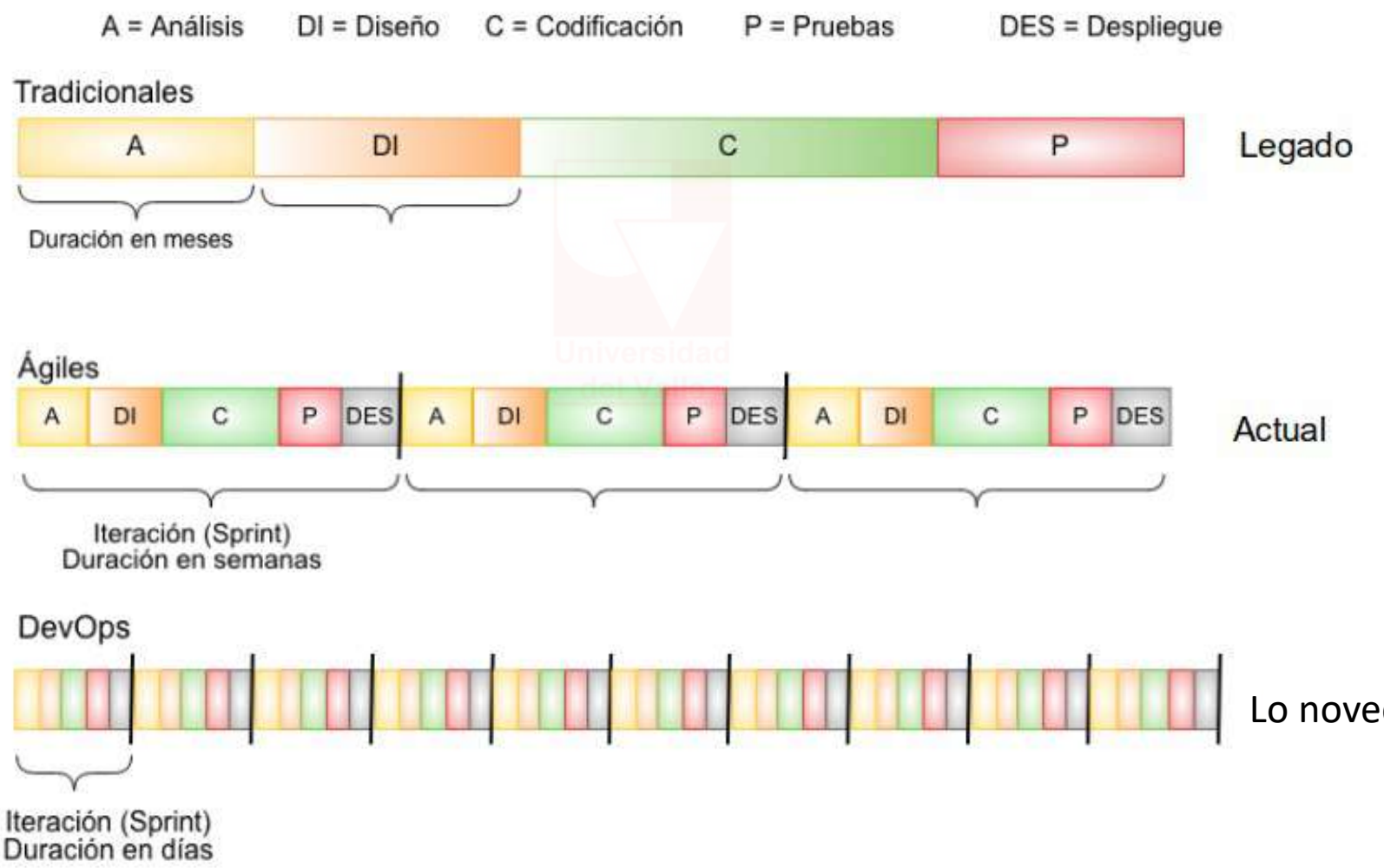


Crisis del software: Problemas de calidad, Entregas tardías y Sobrecostos

Conceptos

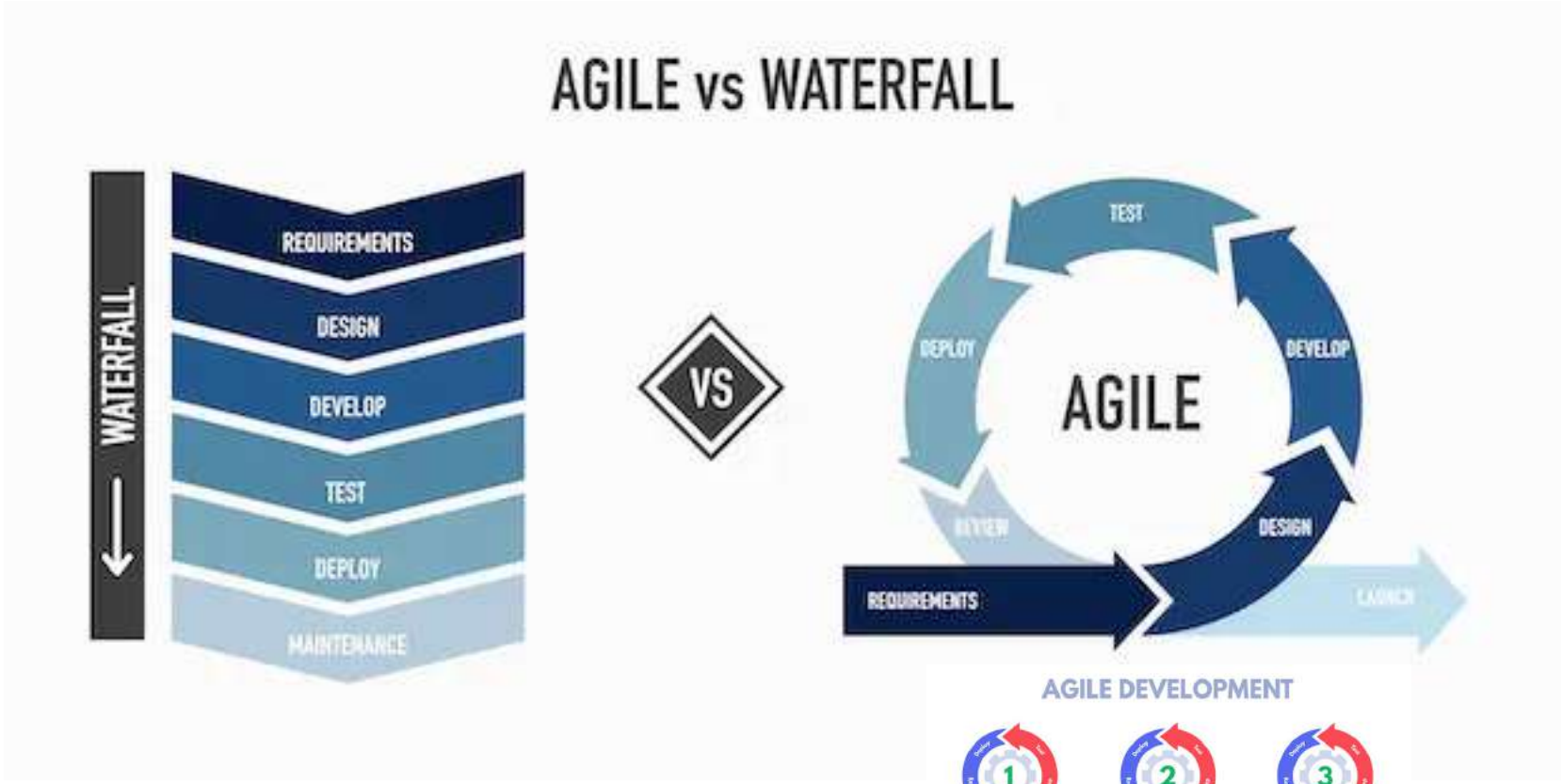
Resumen

Ciclo de vida del desarrollo de software evoluciona



El Modelo de Ciclo de Vida de Desarrollo de software

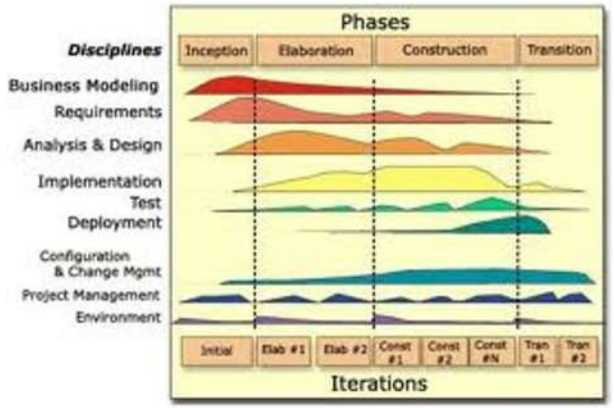
Conjunto de etapas que describen el proceso de desarrollo de software desde su nacimiento hasta su reemplazo o eliminación.

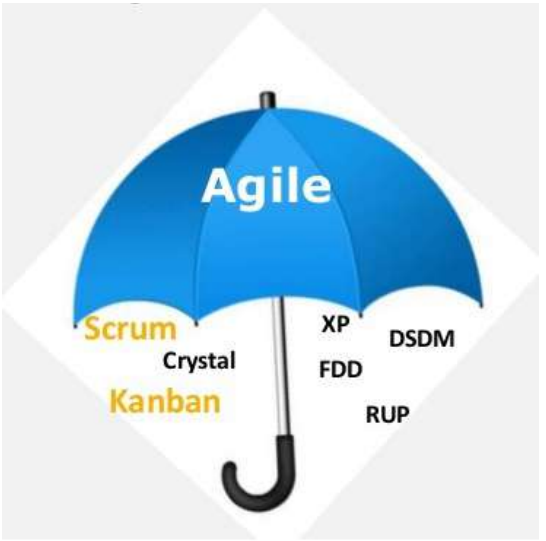


Resumen

Metodologías

- Son una colección de **métodos aplicados a lo largo del ciclo de vida de desarrollo de software**, coherentes entre sí y que siguen una filosofía o enfoque de desarrollo de software.
- Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de un sistema de software.
- Ejemplo: Metodologías Tradicionales: RUP, MSF, ... Metodologías Ágiles: SCRUM, XP, Kanban ...

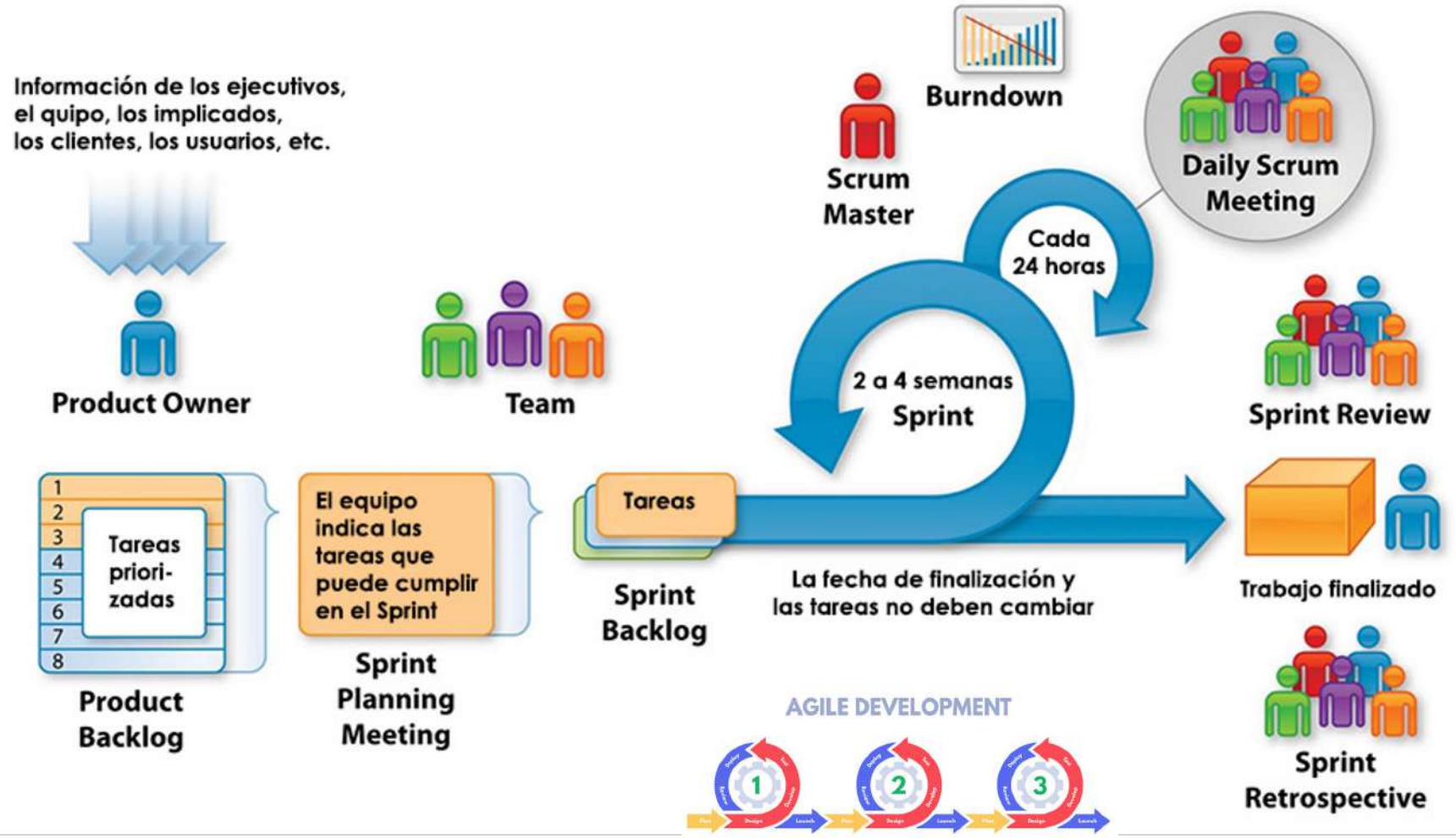




Conceptos

Resumen

Metodología ágil Scrum



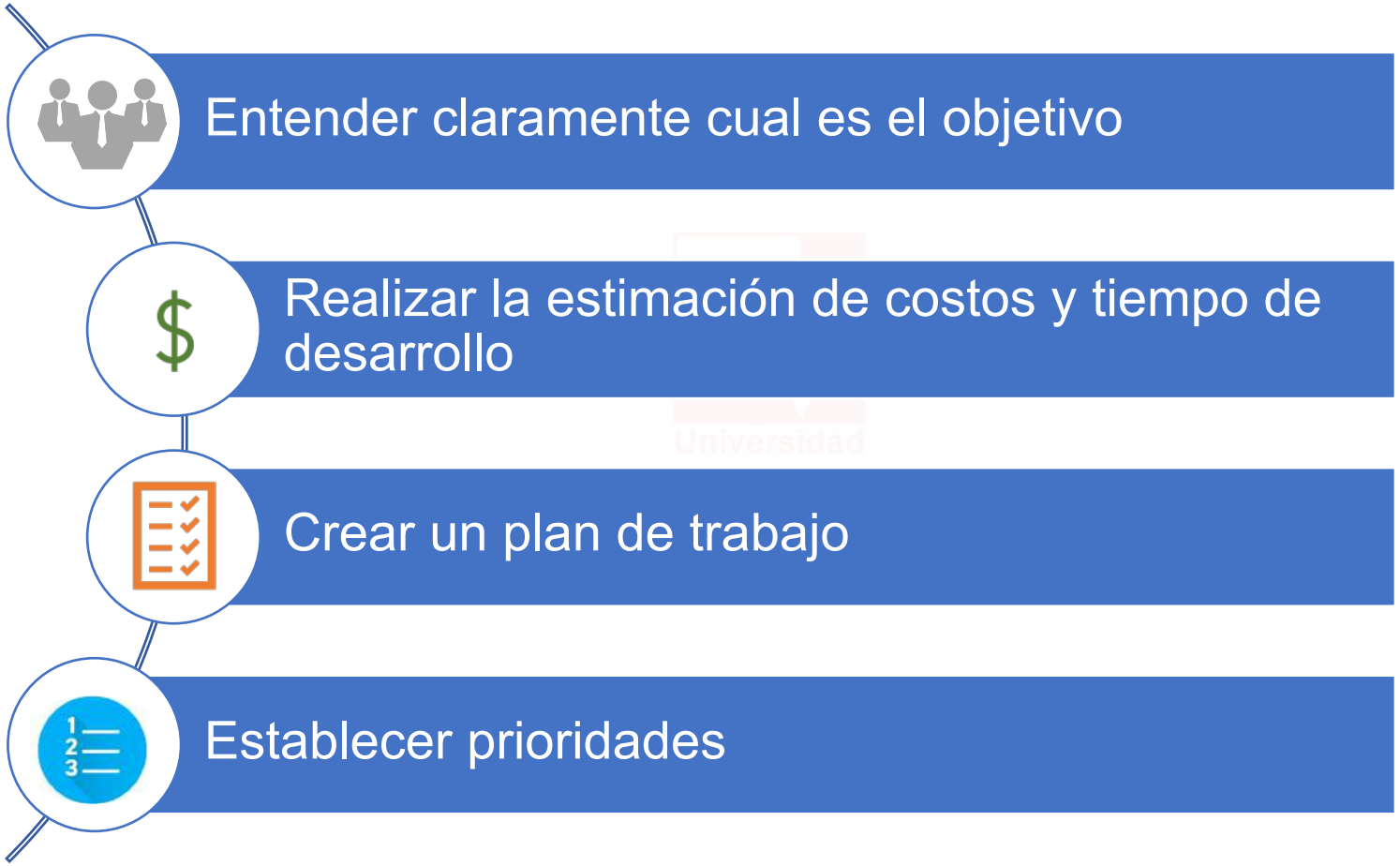
Preguntas ?

REQUERIMIENTOS DEL SOFTWARE

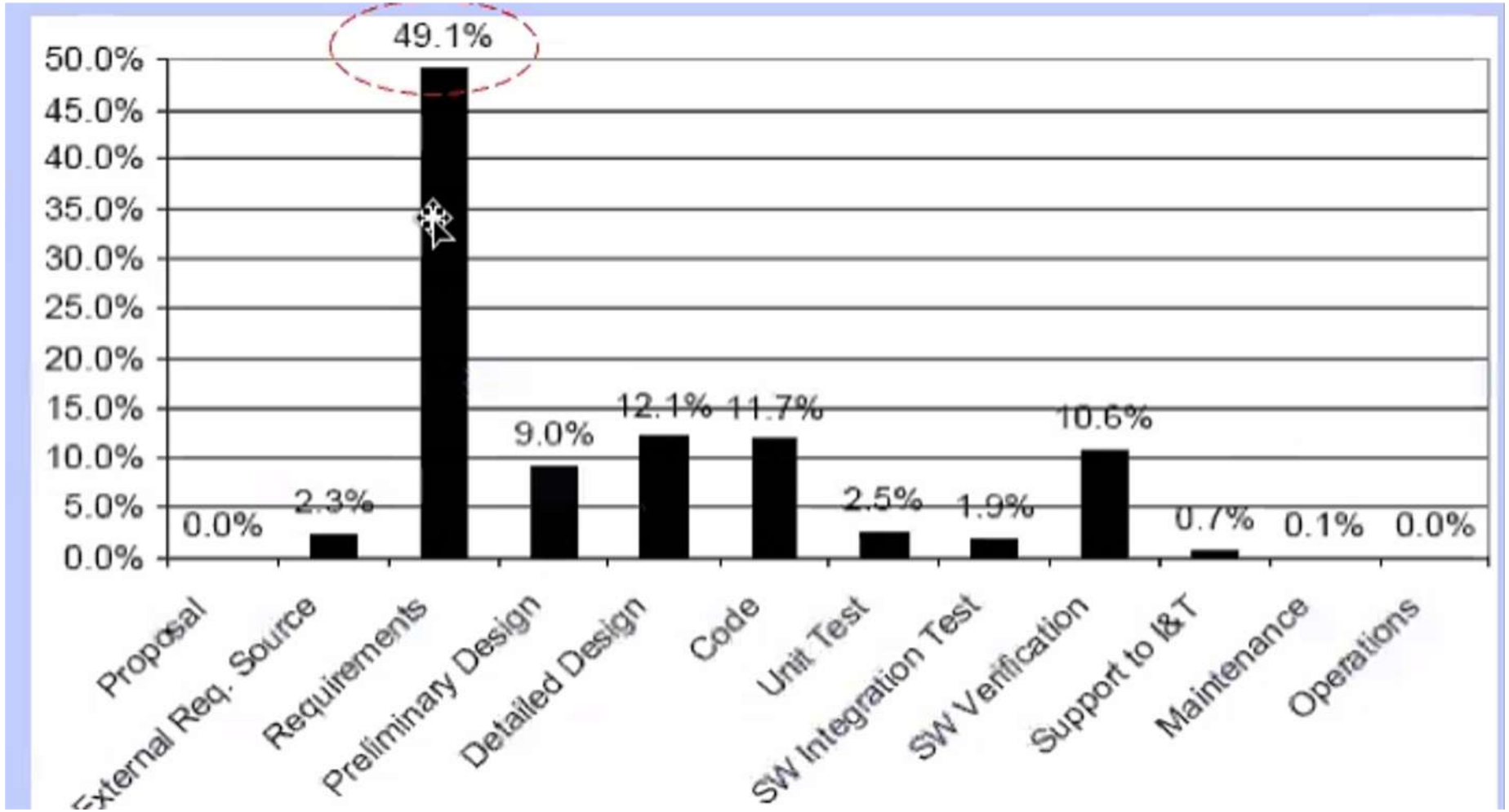
Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambigua, comprobable o medible, y necesaria para la aceptación del producto o proceso.

Los requerimientos especifican lo que desea el cliente y definen las acciones o actividades que debe hacer o cumplir un sistema de software.

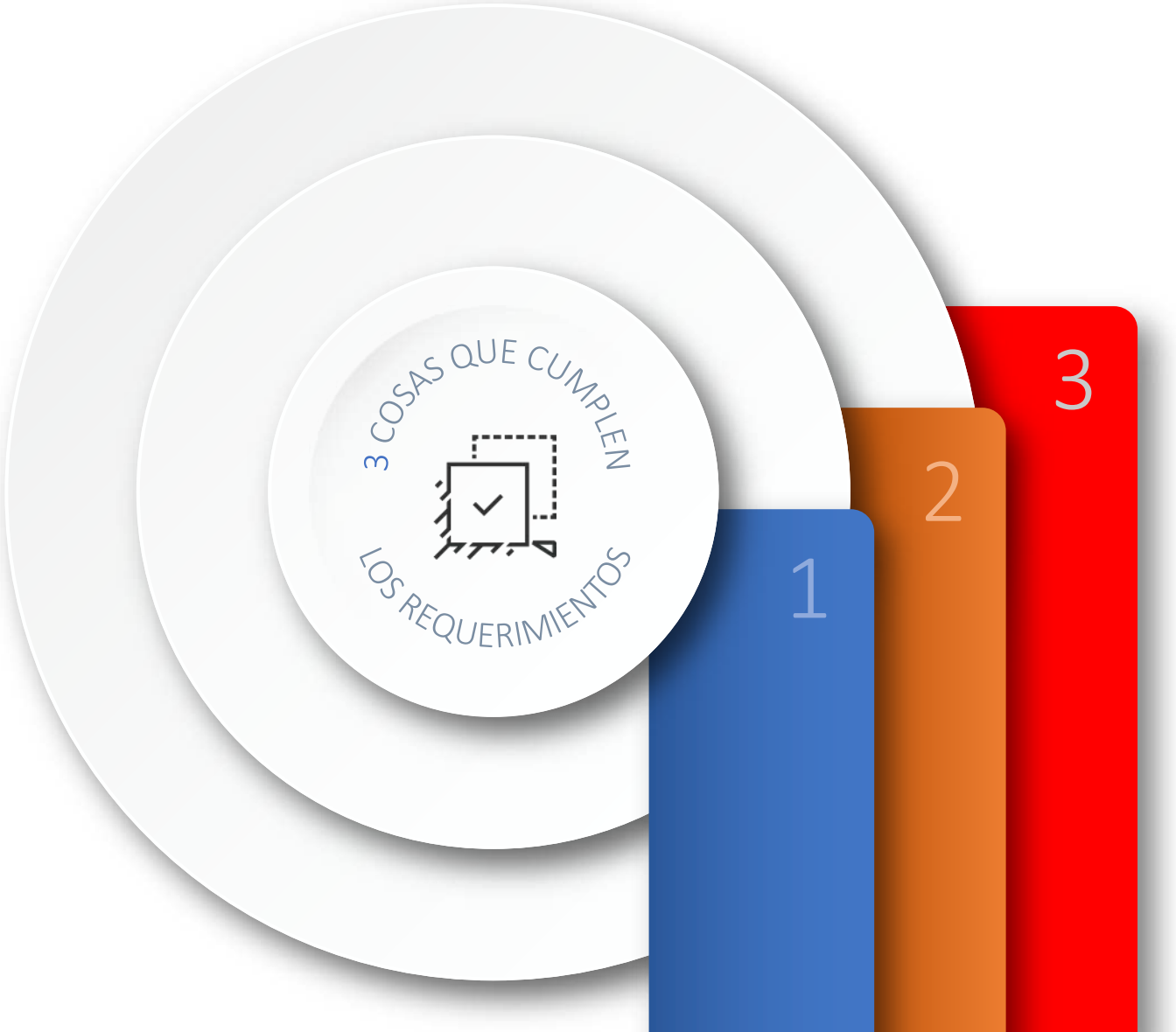
Definir adecuadamente los requerimientos permitirá:



¿De donde provienen los errores en el Desarrollo de software?



¿Qué es un requerimiento?



Resuelve una necesidad

Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

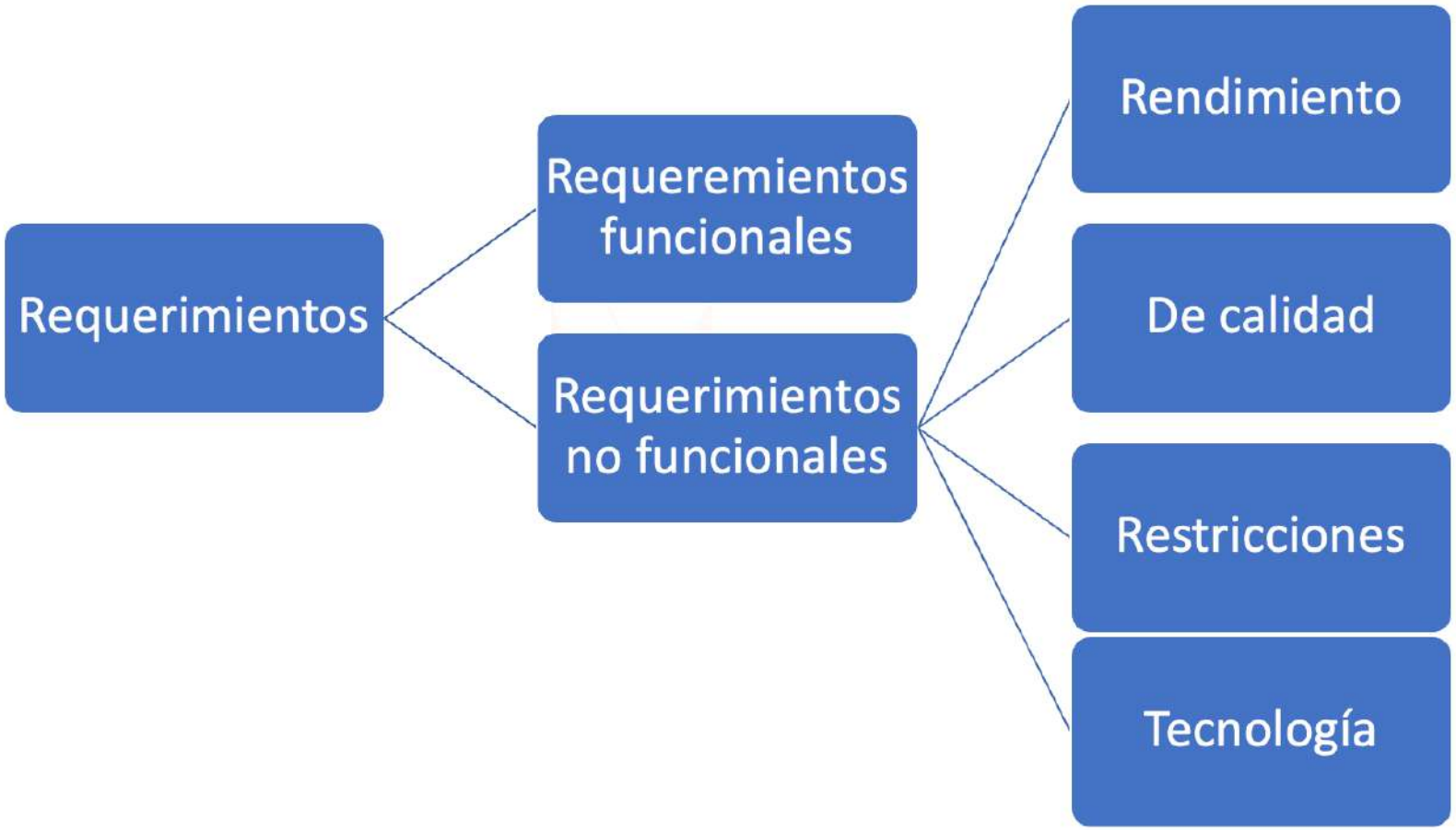
Satisface condiciones

Capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, un estándar, una especificación u otro documento formal.

Define lo que se necesita

Definen lo que el sistema debe hacer en cuanto a sus procesos, consultas, reportes, ...

Clasificación de los requerimientos.



Clasificación de los requerimientos.

Los requerimientos funcionales

Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la forma en que el sistema debe comportarse en situaciones particulares.

Requerimientos no funcionales

Limitaciones en los servicios o funciones ofrecidas por el sistema como de tiempo, limitaciones en el proceso de desarrollo, de rendimiento, de calidad, normas, tecnología, etc

Requerimientos funcionales.

Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos.

Generalmente, los requerimientos funcionales describen el comportamiento del sistema en condiciones específicas.

En algunos casos, los requerimientos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer.

Requerimientos funcionales.

Los requerimientos.

se suelen especificar en lenguaje natural,

se expresan de forma individual y

se organizan de forma jerárquica (a distintos niveles de detalle),

a menudo, se numeran (para facilitar su gestión)

Requerimientos funcionales.

Requerimientos en las metodologías tradicionales

Ejemplos de requerimientos funcionales.

Descripción de requerimientos

El sistema debe permitir crear un usuario del sistema.

El sistema debe permitir registrar participantes a un evento.

El sistema debe permitir registrar la venta de un artículo.

El sistema debe permitir anular una nota débito.

El sistema debe permitir a los usuarios contratistas poder registrar sus pagos de certificación al sistema de seguridad social junto al valor pagado por cada concepto como son: ARL, Pensión y EPS junto con el número de la planilla que lo soporta.

Imprecisión en los requerimientos.

Imprecisión en los requerimientos.

Los problemas surgen cuando los requerimientos no son declarados con precisión.

Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios.

Deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados.

Características de los requerimientos.

Necesario: Lo que pida un requisito debe ser necesario para el producto.

Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe hacer.

No ambiguo: El texto debe ser claro, preciso y tener una única interpretación posible.

Conciso: Debe redactarse en un lenguaje comprensible por los clientes en lugar de uno de tipo técnico y especializado, aunque aún así debe referenciar los aspectos importantes

Consistente: Ningún requisito debe entrar en conflicto con otro requisito diferente. Asimismo, el lenguaje empleado entre los distintos requisitos debe ser consistente también.

Completo: Los requisitos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle.

Alcanzable: Un requisito debe ser un objetivo realista, posible de ser alcanzado en el tiempo y los recursos disponibles.

Verificable: Se debe poder verificar con absoluta certeza, si el requisito fue satisfecho o no. Esta verificación puede lograrse mediante inspección, análisis, demostración o testeo.

Requerimientos no adecuados.

El sistema será lo más fácil de utilizar posible.

El sistema proporcionará una respuesta rápida al usuario.

El sistema se recuperará automáticamente tras producirse un fallo.

Requerimientos adecuados.

Cuando haya hasta 100 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 5 segundos.

Ante un fallo en el software del sistema, no se tardará más de 10 minutos en restaurar los datos del sistema (en un estado válido) y volver a poner en marcha el sistema.

Condiciones verificables

Recomendaciones para redactar requerimientos.

Usar un formato estándar y asegurar la adherencia al mismo para todos los requerimientos.

Utilizar el lenguaje de manera consistente.

Resaltar el texto para distinguir las partes clave del requerimiento.

Evite el uso de jerga informática.

Un requerimiento se especifica en dos pasos:

1. Realizar una **descripción** del requerimiento lo más precisa posible
2. Realizar la **especificación de los detalles** del requerimiento (requerida al momento de implementar)

Formato para registrar detalles de un requerimiento.

- Detalles de los requerimientos que le sirven al desarrollador para entender con más precisión lo que hay que programar.

Descripción	El sistema debe permitir registrar participantes a un evento
Identificador	**Rq-02**
Tipo de Requerimiento	**Tipo de requerimiento:** Funcional
Datos de Entrada	**Nombre, cédula, edad, género, nivel de formación, correo electrónico, evento**
Proceso	Dado los datos de entrada el sistema debe permitir almacenar cada uno de los datos ingresados y enlazar a un participante con el evento al cual desea asistir.
Datos de salida	**Mensaje: "El participante se registró con éxito"**
Resultados esperados	El sistema tendrá un nuevo participante en uno de sus eventos en caso de que el proceso de registro sea exitoso.
Origen	**Necesidades del cliente.**
Dirigido a	**Operadores.**
Prioridad	**5**
Requerimientos asociados	**Rq-01**

Formato para registrar detalles de un requerimiento.

Descripción: **El sistema debe permitir la actualización de los password de los usuarios**	
Identificador: RF003	Tipo de requrimiento: Funcional
Datos de entrada	Login, password actual, nuevo password
Fuente	Formulario de ingreso de datos
Salida	Confirmación en pantalla
Destino	Base de datos
Restricciones	Mínimo 8 caracteres
Proceso	El administrador del sistema tendrá una opción que le permitirá Administrar los usuarios para la modificación de su password, por medio de un formulario ya definido en el cual se solicitara login de identificación y el password actual y el nuevo password. Al confirmar la operación su nuevo password se almacenara en la base de datos y el usuario lo verificara en su próximo ingreso.

Ejercicio

El sistema debe permitir crear un usuario del sistema.

Elaborar la descripción de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad.

Proceso para especificar requerimientos : Sirve para determinar los requerimientos:

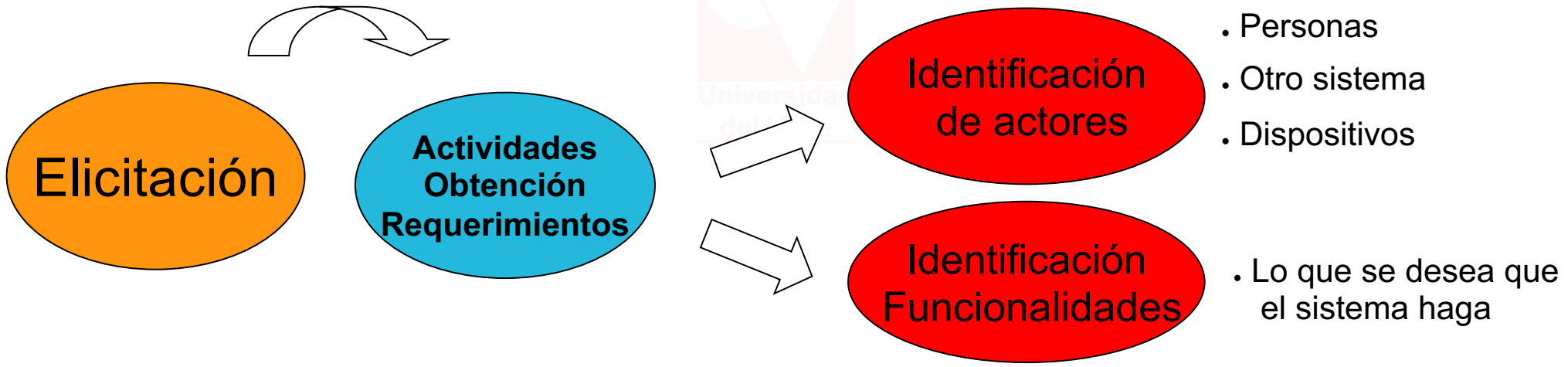
Elicitación: Captura, descubrimiento y adquisición de requerimientos.
(Identificación de actores y funcionalidades)

Análisis de requerimientos: Detectar y resolver conflictos entre los requerimientos, clasificar los requerimientos

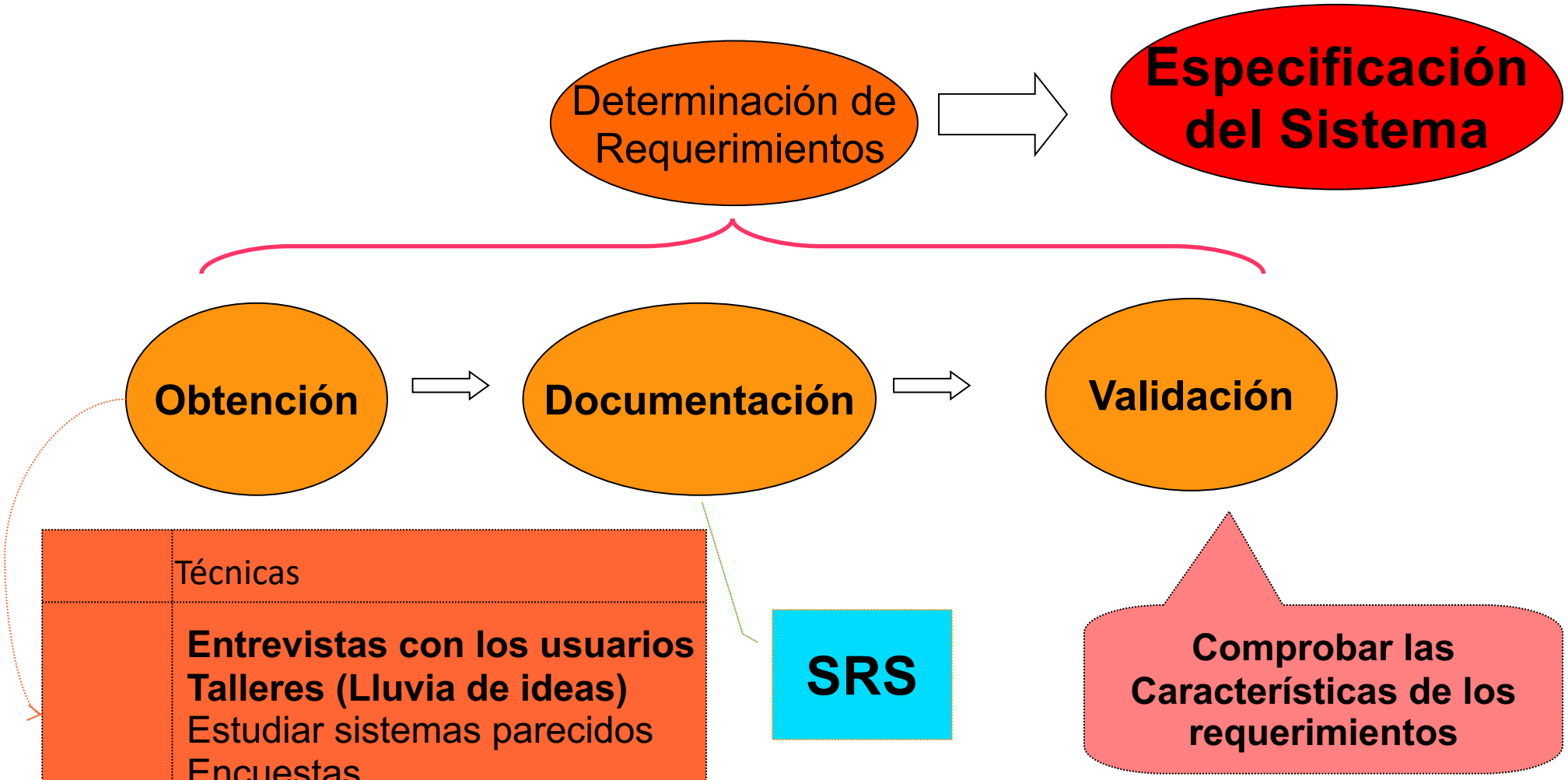
Especificación: Producción del documento de requerimientos

Validación: Validar para asegurar que los requerimientos fueron entendidos y que define lo que el cliente desea

Proceso para especificar requerimientos



Proceso para especificar requerimientos



Requerimientos no funcionales

Técnicas para obtener requerimientos

Entrevistas

Las entrevistas de los usuarios y las partes interesadas son importantes para crear un software adecuado.

La entrevista es un diálogo formal o informal con personas, donde se busca respuesta a un conjunto de preguntas planeadas.

- Identificar el propósito de la entrevista : preguntas que permitan entender con claridad cada cosa que el cliente desea
- Identificar posibles entrevistados : funcionarios de diferentes niveles
- Estudiar el problema planteado
- Familiarizarse con el vocabulario del negocio
- Tomar apuntes o grabar la reunión.

Requerimientos no funcionales

Técnicas para obtener requerimientos

Entrevistas con los usuarios:

Recomendaciones: Evitar

Criticar la información emitida por la otra persona	Lenguaje inadecuado (muy técnico, muy formal o informal)
Completar las frases del otro o interrumpir su discurso	Demostrar la falta de preparación en el tema a tratar
Ser arrogante, o dar la impresión de que Usted sabe más que el otro	Corregir el otro, ya sea en alguna información o en su forma de expresarse
No manifestar interés en la información recibida, en el otro o en su problema	La falta de cortesía, amabilidad, contacto visual o puntualidad
Dar la solución antes de escuchar el problema	Lugar, tiempo o duración inadecuada
Dar un paso en falso (broma o comentario inapropiado)	Presentación personal inadecuada (formal / informal)

Requerimientos no funcionales

Técnicas para obtener requerimientos

Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema.

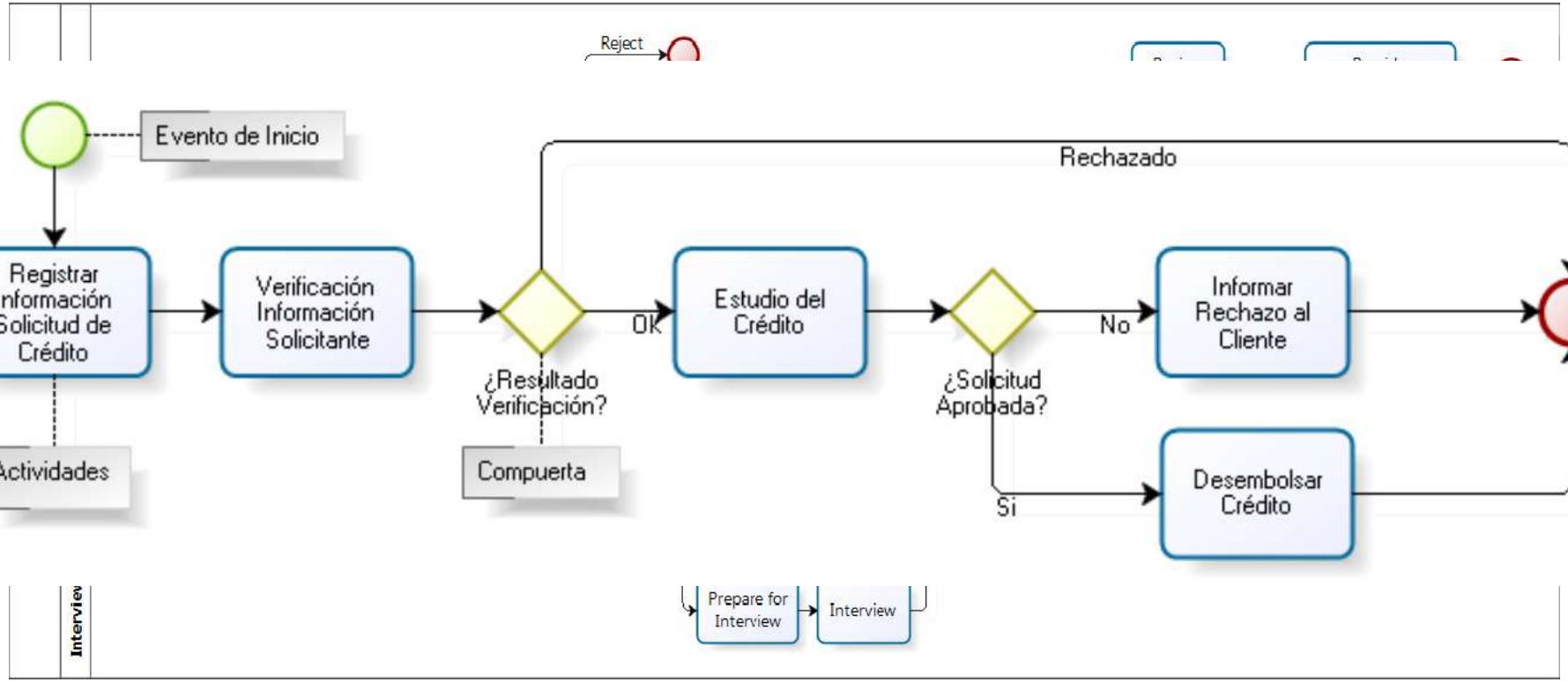
Por lo general, la lluvia de ideas se utiliza para identificar posibles soluciones a los problemas, para aplicaciones nuevas donde hay pocos antecedentes.

- Tener un plan : dividir el problema en partes pequeñas. Ej: hacer diagramas en papel o tableros ayuda a aclarar cosas y capturar colaboración.
- Proponer varias alternativas (ir preparado).
- Combinar ideas.
- Tenga un moderador y limite el tiempo de discusión de cada parte.
- Tomar apuntes o grabar la reunión.

Requerimientos no funcionales

Técnicas para obtener requerimientos

BPM: Business Process Model. **BPMN** Business Process Model Notation



Requerimientos no funcionales

Técnicas para obtener requerimientos

Tarea: Próxima clase control de lectura.

1. Ver video: https://youtu.be/2KZKMY75cJM

2. Leer documento: **Requirements Gathering Techniques**
   https://www.linkedin.com/pulse/requirements-gathering-techniques-samgra-malik/

Requerimientos no funcionales

Técnicas para validación de requerimientos

1. **Validación de expertos**

Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento

2. **Prototipado de interfaz de usuario**

El prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario

Los dos tipos principales de prototipos de interfaz de usuario son:

- **Desechables**: se utilizan sólo para la validación de los requisitos y posteriormente se desechan. Pueden ser prototipos en papel o en software.
- **Evolutivos:** una vez utilizados para la validación de los requisitos, se mejora su calidad y se convierten progresivamente en el producto final.

3. **Recorrido de BPM**: Hacer un BPM que muestre todo el proceso, algoritmo de alto nivel

Requerimientos no funcionales en las metodologías tradicionales

Requerimientos no funcionales

Especifican "qué tan bien" y "como" debe comportarse un sistema

Imponen **restricciones** que típicamente limitan los requerimientos funcionales

También conocidos como "requisitos técnicos", "atributos de calidad "o" requisitos de calidad de servicio "

Requerimientos no funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

De forma alternativa, definen las **restricciones** del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Factores internos: Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las *políticas* de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware.

Factores externos: como los reglamentos de seguridad, las políticas del govierno, entre otros.

Tipos de requerimientos no funcionales

Eficiencia:

- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema debe ser capaz de operar adecuadamente con hasta 100 usuarios con sesiones concurrentes.

Seguridad de lógica y de datos

- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador del sistema.
- Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.

Usabilidad

- El sistema debe contar con manuales de usuario estructurados por cada módulo y funcionalidad.
- El sistema debe contar con un módulo de ayuda en línea.

Tecnología

- El sistema debe ser desarrollado usando el lenguaje JavaPython 3.11.2
- El sistema debe funcionar en el sistema operativo Linux, Android, Windows y MacO

Requerimientos funcionales.

Requerimientos en las metodologías ágiles

Requerimientos en las metodologías ágiles

El Desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles.

- Las metodologías ágiles se concibieron como una alternativa a las prácticas de desarrollo de software tradicional

- Es una “sombrilla” para un conjunto de valores, principios y prácticas

- El desarrollo ágil es una forma diferente de gestionar equipos y proyectos de desarrollo de Software

- **Las Historias de Usuario (HU) es el mecanismo usado para obtener las necesidades del usuario.**

Requerimientos en las metodologías ágiles

En el desarrollo ágil de software, las historias de usuario ayudan a articular el **valor que puede aportar** una característica de la aplicación a desarrollar y a comprender mejor por qué los usuarios quieren una determinada funcionalidad.

Técnicas para obtener requerimientos en las metodologías ágiles

	Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudiar sistemas parecidos Encuestas Prototipos BPM …

https://www.linkedin.com/pulse/requirements-gathering-techniques-samgra-malik/

Requerimientos en las metodologías ágiles

Historias de usuario

Las Historias de Usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software.

Una historia debe permitir conocer la base de las necesidades del usuario, así como los detalles que después permitan fijar la batería de pruebas para poder determinar si un desarrollo es correcto o no.

Las historias de usuario luego se pueden dividir en tareas técnicas para su implementación.

Las historias de usuario son la base para aplicar las metodologías de desarrollo.

Requerimientos en las metodologías ágiles

Características deseables de las historias de usuario (HU)

El modelo INVEST

Una buena historia de usuario también sigue el modelo de INVEST: Independiente, Negociable, Estimable, Pequeña (Small), y Testeable. Veamos lo que significa.

- **Independiente** - una historia debería ser independiente de otras. Facilitan la planificacion, priorizar y estimación.
- **Negociable** - La "tarjeta" de la historia es tan sólo una descripción corta que no incluye detalles. Los detalles se añaden mediante la conversación.
- **Valiosa** - cada historia tiene que tener valor para el cliente (para el usuario o para el comprador).
- **Estimable** - el equipo necesitan poder estimar una historia de usuario. Historias demasiado grandes o inconcretas, no se pueden estimar.
- **Pequeña** - una buena historia debe ser pequeña en esfuerzo, debería ser realizable en menos de una semana.
- **Testeable** - una historia necesita poder probarse y saber que la HU se ha completado con éxito.

Requerimientos en las metodologías ágiles

Ejemplos de historias de usuario

Como estudiante deseo registrarme en un curso para matricularme en la universidad

Actor — La tarea — Propósito

Plantilla: Como <Actor> deseo <tarea> para <propósito>

Requerimientos en las metodologías ágiles

Ejemplos de historias de usuario

1. El sistema debe permitir crear un usuario del sistema.

2. Como Vendedor, deseo registrar los productos y cantidades que me solicita un cliente para crear un pedido de venta.

3. Como Auxiliar, deseo poder registrar los participantes a un evento para conocer cuantas personas están inscritas al evento.

Ejercicio: Elaborar la descripción de una historia de usuario

Plantilla: Como <Actor> deseo <tarea> para <propósito>

Elaborar la descripción de una historia de usuario para crear una factura de venta de un producto a un cliente

Elaborar la descripción de una historia para el proceso de retiro en efectivo en un cajero automático

Características de la historia de usuario

Ser lo suficientemente completa como para demostrar el valor para el usuario.

Centrarse en el usuario.

Si no se identifica necesidades precisas, se puede iniciar con una historia de usuario **Épica**.

Ser breve, sencilla y clara.

Spyke (investigar de cosas que el equipo de desarrollo no conoce)

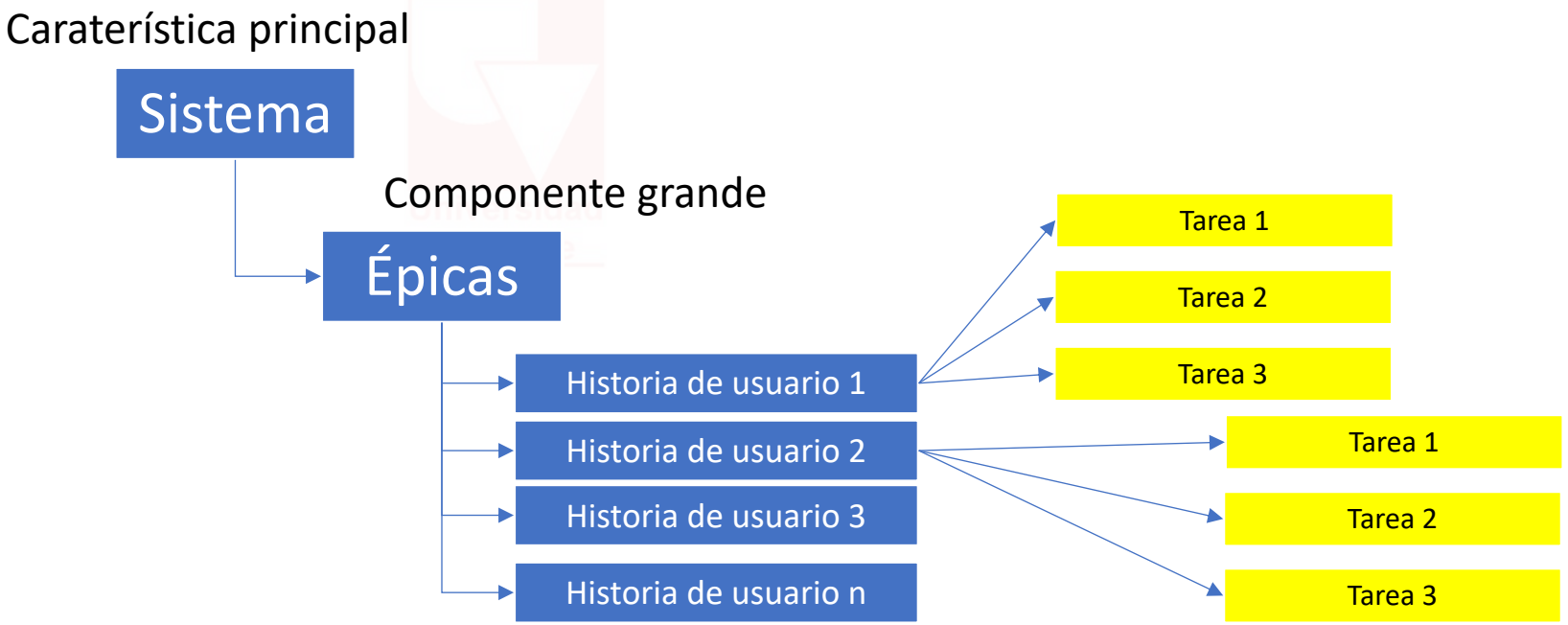
Incluir archivos y documentación de apoyo si es necesario.

Ser lo suficientemente completo como para demostrar su valor, pero lo suficientemente sencillo como para desarrollarlo en una sola iteración.

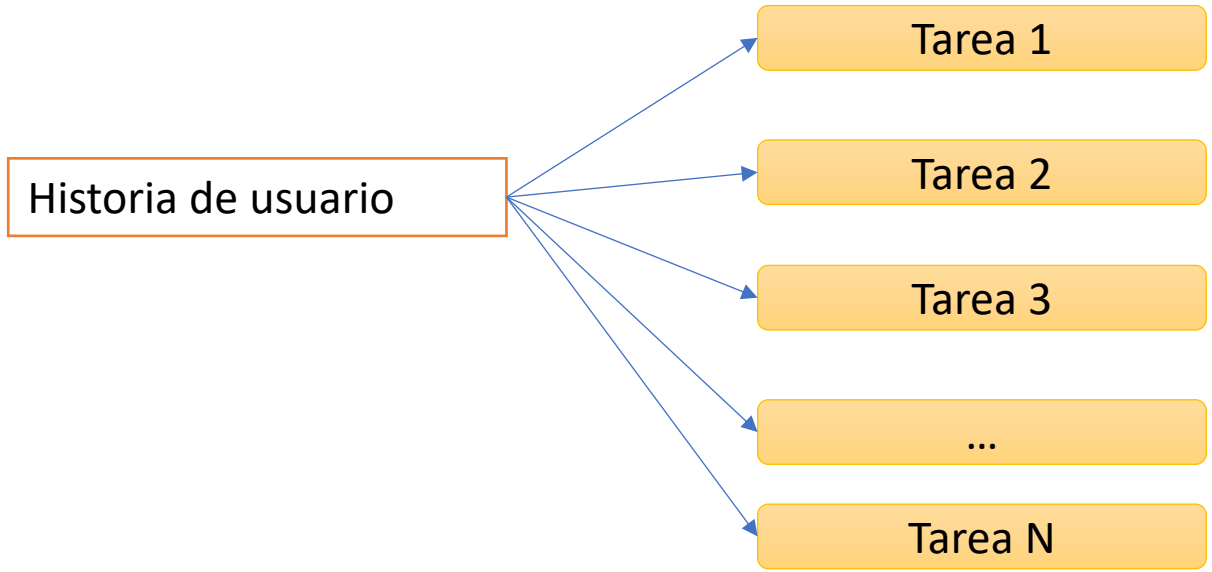
Se recomienda **dividir una historia de usuario en tareas** sencillas

Historia de usuario épicas

"Una historia épica es una gran historia de usuario que no se puede entregar como se define en una sola iteración o es lo suficientemente grande como para dividirse en historias de usuario más pequeñas."



Descomponer una Historia de usuario en tareas



¿Para qué dividir una historia de usuario en tareas ?

- Para expresar los pasos para implementar una historia de usuario.
- Para precisar la actividad en la que esta trabajando.
- Puede ayudar para pedir colaboración mas específicas.
- Puede conducir a estimaciones más precisas.
- Se pueden usar como una medida de progreso.

Descomponer una Historia de usuario en tareas

¿Cómo descomponer historia de usuario en tareas?

No hay una formula para hacer esto.

Depende de la HU las tareas a realizar.

Recomendaciones:

- Descomponga la HU en tareas técnicas.
- No programe tareas demasido pequeñas.
- Trate de programar tareas con duración de horas y que puedan mostrar avances.
- Antes de implementar tareas debes de cero verifique si hay componentes listos que pueda usar.

Descomponer una Historia de usuario en tareas

Ejemplo: HU003. Como usuario operador deseo poder registrar en el sistema datos de un cliente para crear una cuenta al cliente.

¿Cómo podemos dividir esta HU en tareas?

Dividir la HU en tareas técnicas

Tarea	Descripción de la tarea
1	Diseñar e Implementar formulario para capturar los datos del cliente.
2	Adicionar tablas y/o campos en la BDatos para registrar los datos del formulario.
3	Implementar lógica para leer y validar los campos del formulario.
4	Implementar mensaje fallo del proceso de lectura o validación si ocurre
5	Implementar el registro de datos del formulario en la BDatos.
6	Implementar mensaje de éxito o fallo del proceso.
7	Ejecutar plan de pruebas (Realizar las pruebas de aceptación definidas).
8	Realizar actividades del DoD.

Historia de usuario épicas

Como usuario administrador del sistema deseo poder gestionar usuarios para poder administrar los usuarios

Esta épica se puede subdividir en:

- Como usuario administrador deseo poder **crear** un usuario para registralo en el sistema.
- Como usuario administrador deseo poder **modificar** los datos de un usuario para actualizar información de un usuario.
- Como usuario administrador deseo poder **listar** los usuarios registrados en el sistema para conocer el tipo de usuarios se tienen el sistema.
- Como usuario administrador deseo inhabilitar (**borrar**) un usuario para desactivar un usuario del sistema.

Historia de usuario épicas

Como Vicepresidente de mercadeo y ventas, quiero revisar el desempeño histórico de las ventas, para poder identificar las regiones geográficas y productos de mejor desempeño

Esta épica se puede subdividir en:

- Como VP de Mercadeo, deseo seleccionar el período de tiempo en el cual realizaré la revisión de las ventas para poder revisar las ventas de diferentes periodos.
- Como VP de Mercadeo, deseo clasificar la información de ventas por región geográfica para tener la información clasificada por regiones.
- Como VP de Mercadeo, deseo clasificar la información de ventas por productos para tener la información clasificada según el tipo de producto.

Ejemplos de HU:

http://www.pmoinformatica.com/2015/05/historias-de-usuario-ejemplos.html

Detalles de una historia de usuario

Historia de Usuario (HU)			
Código HU:	HU0023	Fecha:	17/02/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Gerente	Puntos:	3

1

2 **Descripción:**
Como gerente deseo ver las ventas semanales de mi empresa para conocer como se comportan las ventas día a día.

3 **Detalles de la HU**:[Colocar aquí toda la información que se requiera para entender la HU: **Tareas**, Texto (entradas, proceso, salida), fotos videos, audio, BPM, diseño de interfaces de usuario, etc]

4 **Restricciones:**
1. Solo los usuarios con privilegios de gerente pueden acceder al reporte.
2. El tiempo de despliegue del reporte no debe exceder 4 segundos

5 **Criterios de aceptación: (Detalle de las HU desde el punto de vista de calidad y se traducen en pruebas)**
Criterio 1, Criterio 2, …

6 **DoD (Definition of Done):**
Lista de actividades que se deben cumplir para que la historia esté en donciciones de ser entregado al cliente.

Detalles de una historia de usuario : Criterios de aceptación

Como usuario deseo poder loguearme en el sistema para poder ingresar al sistema

Item	**Criterio de aceptación :** las condiciones que un producto de software debe satisfacer para ser **aceptado** por un usuario, cliente o stakeholder.
1	Un usuario no puede enviar un formulario sin completar todos los campos.
2	El login no puede ser igual al password.
3	El password debe tener al menos 6 caracteres, una letra en mayúscula, un símbolo y un número.
4	Los campos de entrada de datos se deben validar antes de ejecutarlos o guardarlos.

Prueba	Login	Password	Resultado esperado
1			Login o password vacíos
2	123456789	123456789	Login o password incorrecto
3	usuario	Secret()22	Login exitoso
4	prueba	SELECT * FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'	Login o password incorrecto

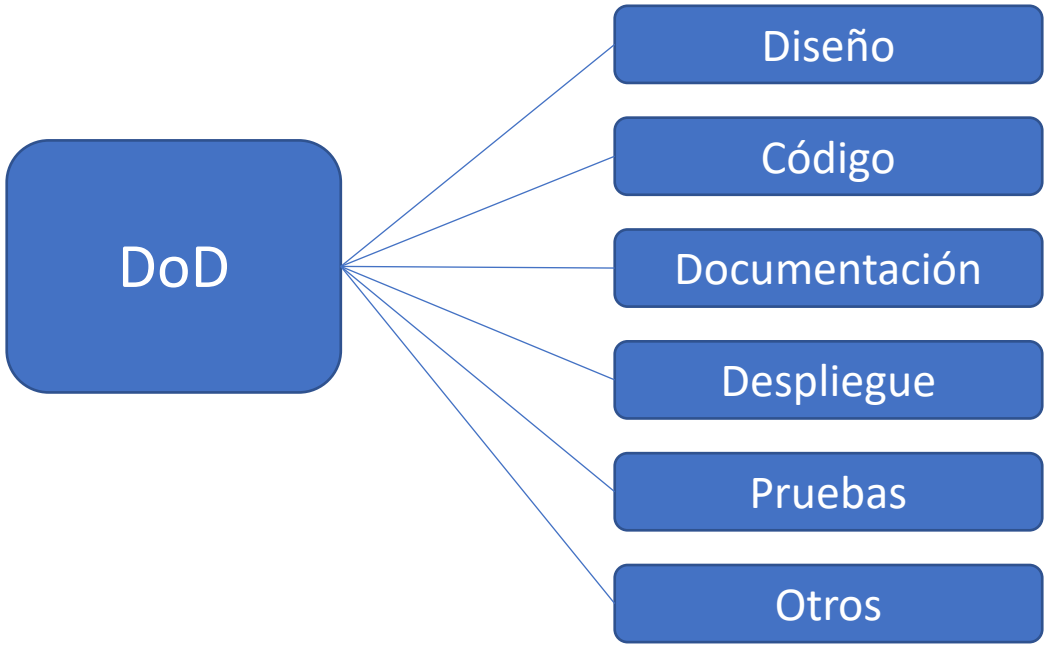
Pruebas de Aceptación

Detalles de una historia de usuario

DoD (Definition of Done): <<Definición de hecho>>

Cada equipo de desarrollo tiene su propia definición, pero la **DoD** puede ser una lista de actividades o simplemente una serie de acuerdos que agregan valor verificable y demostrable al producto.

Es un entendimiento compartido de lo que significa que una historia de usuario está realmente terminada.



Detalles de una historia de usuario

DoD (Definition of Done): <<Definición de hecho>>

DoD (Definition of Done): Ejemplo para cada historia de usuario

1. El diseño de la historia fue aprobada (Product owner y equipo de desarrollo)
2. La historia esta terminada a criterio del desarrollador
3. La historia paso las pruebas según los criterios de aceptación en el ambiente de desarrollo.
4. La historia paso las pruebas definidas en el ambiente de pruebas y de preproducción
5. Se realizó la documentación (técnica y manuales de usuario si se requiere)
6. [Se verifica si satisface el requerimiento y si el código sigue es estándar de codificación acordado.]
7. Esta todo preparado para hacer el despliegue (subirla a producción).
8. La historia esta marcada como terminada en la herramienta de control

Detalles de una historia de usuario

DoD (Definition of Done): <<Definición de hecho>>

DoD para el Sprint: Ejemplo

1. Se cumple con el DoD de todas las Historias del Sprint.
2. Todas las historias acordadas durante en Sprint Planning están *terminadas*.
3. Demo verificado en el Review por parte de Product Owner y *stakeholders.*
4. OK del Product Owner al Sprint.

Detalles de una historia de usuario

DoD (Definition of Done): <<Definición de hecho>>

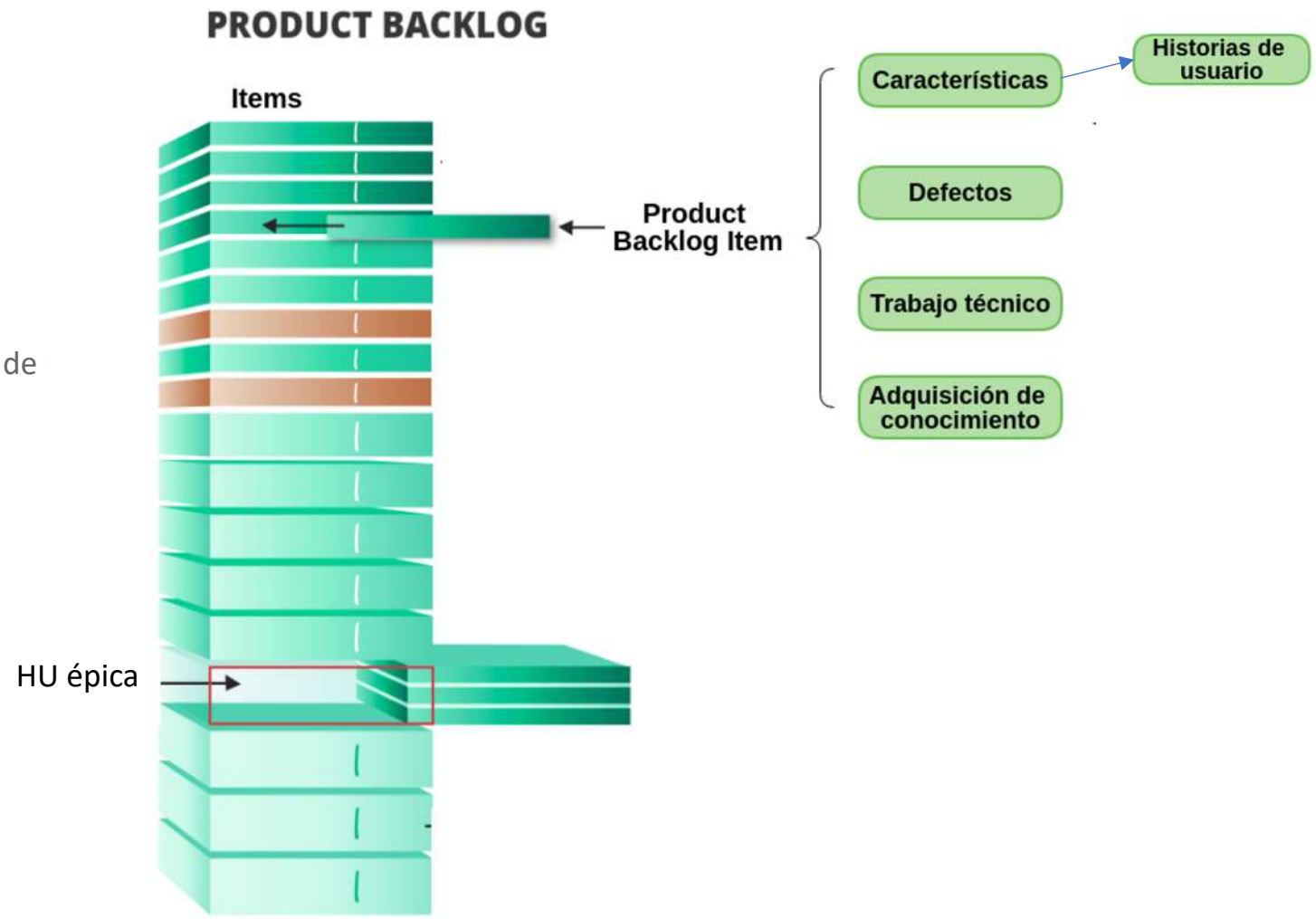
DoD para el Proyecto o cada Release: Ejemplo

1. Se cumple con el *DoD* de cada Sprint del Release o del proyecto.
2. Se cumple con las pruebas definidas para el Release en preproducción (carga, integración, seguridad, etc)
3. Documentación unificada y subida a la herramienta de documentación.

Historia de usuario

Product BackLog

Lista priorizada y estimada de hitorias de usuario



Product Backlog

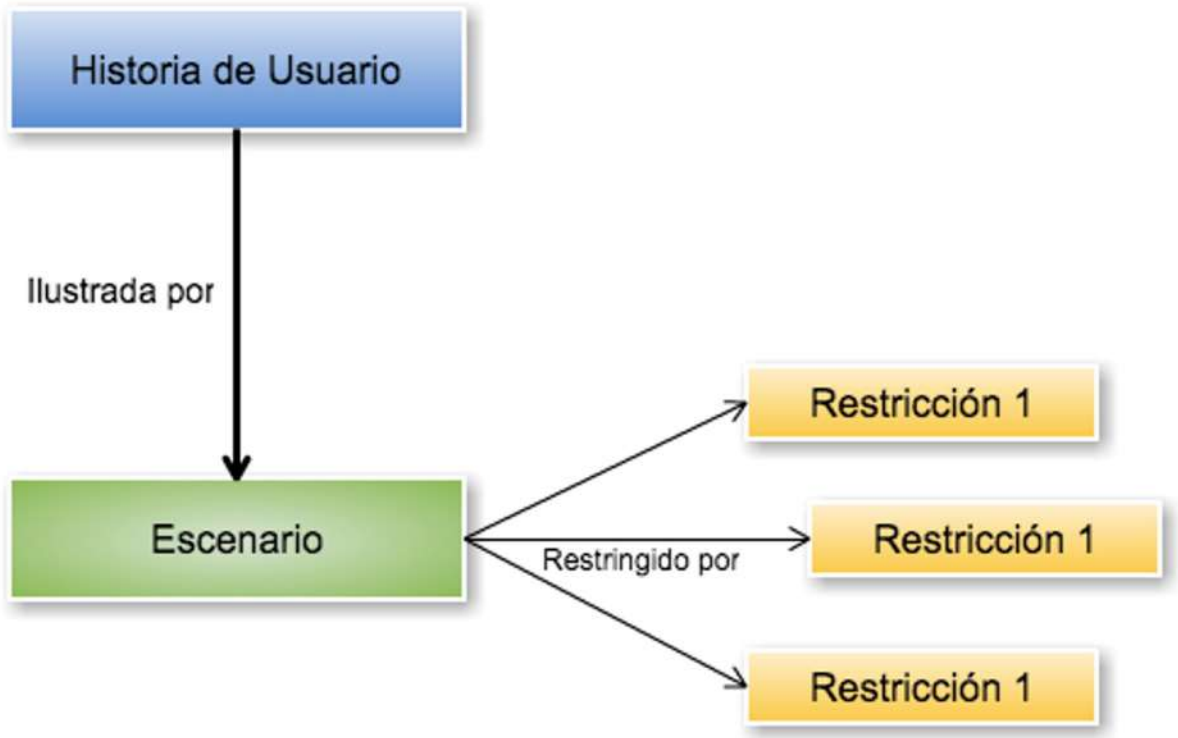
Módulo / Épica	ID	Nombre	Descripción	Prioridad	Estimación
Gestión de usuarios	HU-01	Admin Sistema CREATE	Como administrador del sistema debo poder registrar cualquier tipo de usuario en la aplicación (Subcontratista, o trabajador)	Alta	3
	HU-02	Admin Sistema DELETE	Como administrador del sistema debo poder borrar cualquier tipo de usuario en la aplicación (Subcontratista, o trabajador)	Alta	2
	HU-03	Admin Sistema UPDATE	Como administrador del sistema debo poder modificar cualquier tipo de usuario en la aplicación (Subcontratista, o trabajador)	Alta	2
	HU-04	Admin Sistema READ	Como administrador del sistema debo poder observar (listar) los usuarios registrados en la aplicación (Subcontratista, o trabajador)	Alta	3
	HU-05	Admin Sistema LOGIN	Como administrador del sistema deseo poder iniciar sesión en la aplicación usando mis credenciales	Alta	2
	HU-06	Subcontratista LOGIN	Como subcontratista deseo poder iniciar sesión en la aplicación usando mis credenciales	Alta	2
	HU-07	Trabajador LOGIN	Como trabajador deseo poder iniciar sesión en la aplicación usando mis credenciales	Alta	2
	HU-08	Admin Sistema LOGOUT	Como administrador del sistema deseo poder cerrar sesión en la aplicación	Alta	2
	HU-09	Subcontratista LOGOUT	Como subcontratista deseo poder cerrar sesión en la aplicación	Alta	2
	HU-10	Trabajador LOGOUT	Como trabajador deseo poder cerrar sesión en la aplicación	Alta	2
Gestión de avances de obra	HU-11	Obra STATE READ	Como Subcontratista deseo visualizar los avances de obra publicados por los trabajadores durante el desarrollo de la obra	Alta	4
	HU-12	Obra formulario CREATE	Como Jefe de obra deseo poder registrar mediante formularios el avance de la obra	Media	3
	HU-13	obra audio CREATE	Como técnico, arquitecto residente deseo poder registrar mediante notas de voz el avance de la obra	Media	3
	HU-14	Obra fotos CREATE	Como técnico, arquitecto residente deseo poder registrar mediante fotos el avance de la obra	Media	5
Mapas	HU-15	Mapa READ	Como subcontratista o trabajador deseo poder visualizar en un mapa mis obras en proceso	Media	4
	HU-16	Planos READ	Como subcontratista o trabajador deseo poder visualizar los planos de mis obras en proceso	Media	4
Gestión de Inventarios (almacén)	HU-17	Almacén STATE	Como jefe de almacen deseo conocer el inventario actual de los materiales	Baja	3
	HU-18	Almacén SUPPLY	Como jefe de almacen deseo tener acceso a la lista con los distintos distribuidores	Baja	3
	HU-19	Almacen NOTIFY	Como jefe de almacen deseo recibir una alarma cuando el material este cerca de acabarse (ejemplo solo 10% de disponibilidad) y se requiera más de lo disponible para finalizar la obra.	Baja	3
Flujo de trabajo de obras	HU-20	Solicitud CREATE	Como Jefe de obra, deseo solicitar al Jefe de almacén materiales para desarrollar una obra	Baja	3
	HU-21	Solicitud STATUS	Como Jefe de almacen, deseo aceptar o rechazar una solicitud de materiales por parte de un jefe de obra	Baja	2

Requerimientos no funcionales en las metodologías ágiles

Requerimientos no funcionales en las metodologías ágiles

Imponen restricciones para guiar el trabajo en un proyecto de desarrollo

- Una restricción es una condición para los requisitos en línea con las expectativas de calidad
- Una restricción establece un límite o algo a cumplir



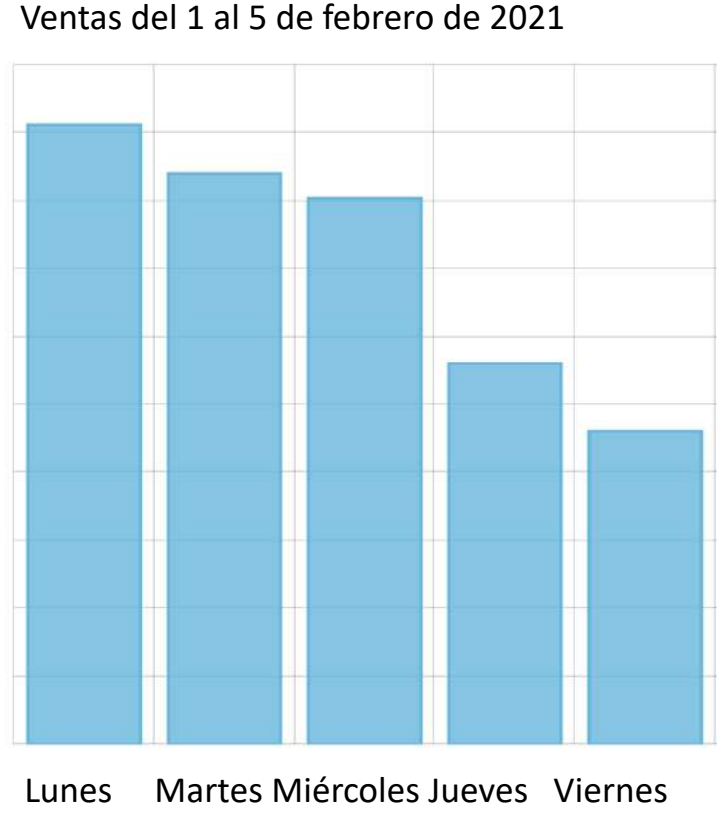
Requerimientos no funcionales en las metodologías ágiles (restricciones)

Ejemplo:

HU: Como gerente deseo ver las ventas de mi empresa para saber como se comportan las ventas día a día.

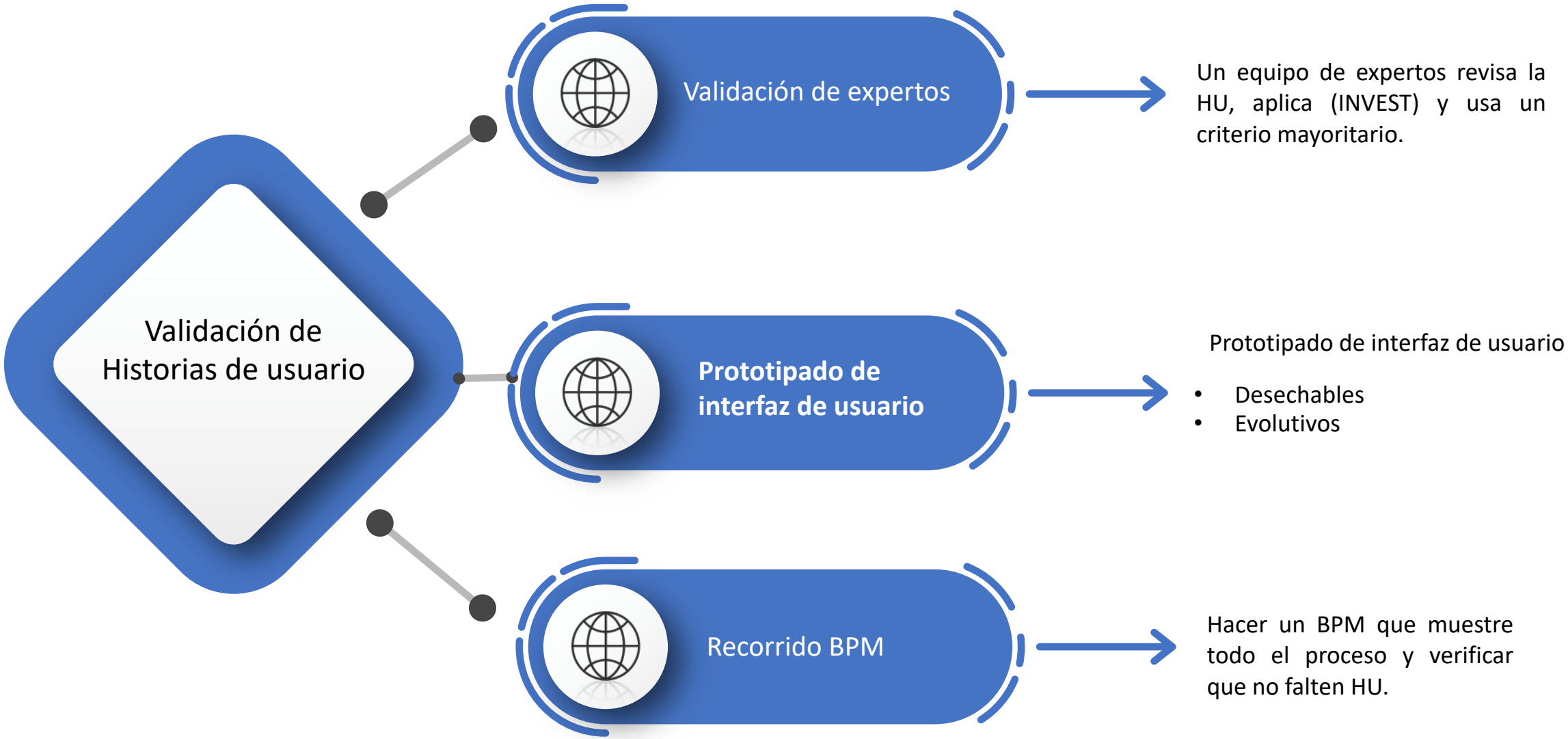
Restricciones:
1. Las ventas se deben desplegar mediante un diagrama de barras
2. El tiempo de despliegue del reporte no debe exceder de 5 segundos

Otros:
- El cuadro de búsqueda debe aceptar valores alfanuméricos
- El botón Buscar debe aparecer debajo del cuadro de búsqueda
- Los resultados de búsqueda deben mostrar solo 15 elementos buscados por página
- El sistema responde a todas las solicitudes de búsqueda dentro de los 5 segundos posteriores a la recepción de la solicitud.



Nota: En muchos casos las restricciones se definen de manera global para el sistema y solo se colocan restricciones en un historia de usuario si es algo muy crítico para el sistema o particular de una historia.

Validación de historias de usuario



Preguntas ?

Actividades

Tarea: Próxima clase control de lectura.

1. Ver video: https://youtu.be/2KZKMY75cJM

2. Leer documento: **Requirements Gathering Techniques**
   https://www.linkedin.com/pulse/requirements-gathering-techniques-samgra-malik/

Gracias

Universidad del Valle