



# Ingeniería de Software

## Historias de Usuario

Historias de usuario

María Antonieta Soto Ch.

2019



1

## Definición y características

- Una HdU es una corta declaración de intención que describe algo que el sistema necesita hacer para el usuario.
- Se describe desde el punto de vista de quien usará la funcionalidad.
- Características:
  - Cortas y fáciles de leer, entendibles por los desarrolladores, interesados y usuarios.
  - Representan incrementos pequeños de funcionalidad valorada que puede ser desarrollada (y validada) en un sprint (preferiblemente en 3 días).
  - Son relativamente fáciles de estimar porque el esfuerzo de implementarlas se puede determinar rápidamente.
  - No se llevan en documentos, sino más bien en listas organizadas que pueden ordenarse fácilmente.

2

2

## Definición y características

- Características:
  - No se detallan al principio del proyecto, sino que se elaboran sobre una base JIT (just in time).
  - Necesita poco o ningún mantenimiento y se pueden desechar con seguridad después de la implementación.
  - Las HdU, y el código que se crea rápidamente a continuación, sirven como insumo para la documentación, la cual también es elaborada de manera incremental.

3

3

## Plantilla

- **Como** <rol/¿quién?>  
**quiero/deseo/necesito** <funcionalidad/¿qué?>  
**para** <beneficio de negocio/¿por qué?>
- Ejemplos:
  - **Como** miembro de la red social **quiero** ingresar a la red **para** compartir un enlace
  - **Como** cliente **deseo** saber cuáles son las películas disponibles en un teatro cerca de mi casa **para** poder elegir qué película ir a ver el día de hoy

4

4

## Criterios de aceptación

- Los criterios de aceptación nos permiten conocer cuándo la historia está Terminada (criterios de Done de Scrum), de esta forma el equipo de desarrollo no se pierde, pues todo su desarrollo está orientado al resultado esperado, o mejor, orientado a los casos de prueba más que a la especificación

5

5

## Criterios de aceptación Formato BDD

- **Dado** <contexto> [**y** <contexto>...]  
**Cuando** <evento> [**y** <evento>...]  
**Entonces** <resultado / comportamiento  
esperado>  
[**y** <resultado>...]

BDD: Desarrollo conducido por comportamiento (Behavior Driven Development)

6

6

## Ejemplo

- Considerando la HdU:
  - Como cliente **deseo** saber cuáles son las películas disponibles en un teatro cerca de mi casa **para** poder elegir qué película ir a ver el día de hoy.
- Escenario 1: listar de a 10 películas
  - **Dado** que me encuentro en cualquier sección de la aplicación
  - **Cuando** selecciono ver listado de películas
  - **Y** antes no se ha seleccionado un tamaño de lista diferente
  - **Entonces** el sistema me listará 10 películas
  - **Y** estas aparecerán en orden de estreno con su respectiva imagen y una descripción.

7

7

## Ejemplo

- Como dueño de una tarjeta quiero retirar efectivo de un cajero automático de manera que pueda tener dinero cuando el banco esté cerrado.
- Escenario 1: la cuenta tiene suficientes fondos
  - Dado que el saldo de la cuenta es \$100
  - Y la tarjeta es válida
  - Y el cajero tiene suficiente dinero
  - Cuando el dueño de la cuenta solicita \$20
  - Entonces el cajero debe dispensar \$20
  - Y el saldo de la cuenta debe ser \$180
  - Y la tarjeta debe ser retornada

8

8

# Ejemplo

- Escenario 2: la cuenta no tiene suficientes fondos  
Dado que el saldo de la cuenta es \$10  
Y la tarjeta es válida  
Y el cajero tiene suficiente dinero  
Cuando el dueño de la cuenta solicita \$20  
Entonces el cajero no debe dispensar dinero  
Y el cajero debe decir que no hay suficientes fondos  
Y el saldo de la cuenta debe ser \$10  
Y la tarjeta debe ser retornada

9

9

# Ejemplo

Como **Cliente del Banco**  
Quiero **Ingresar mi información laboral actual**  
Para **Que el Banco determine si me puede hacer un préstamo o no**

Criterios de aceptación (se pueden manejar dos opciones) Criterios de aceptación en Prosa	Criterios de Aceptación en Formato BDD GIVEN – DADO WHEN – CUANDO THEN – ENTONCES (Este forma ayuda a que no queden cabos sueltos en la historia)
Que solicite lo datos de la empresa	<b>Escenario 1: Solicitar Datos de la empresa</b> <b>DADO</b> que me encuentro en la página de cliente <b>Y</b> se haya registrado al cliente como empleado <b>CUANDO</b> se soliciten los datos de la empresa <b>ENTONCES</b> se pedirá el nombre de la empresa.
Que solicite el NIT (Número de Identificación Tributaria) para las empresas) y lo valide	<b>Escenario 2: Solicitud de NIT</b> <b>DADO</b> que me encuentro en la página de cliente <b>Y</b> se haya registrado al cliente como empleado <b>CUANDO</b> solicite la información de NIT <b>ENTONCES</b> se verificará el número del NIT que su estructura sea correcta.

10

10

## Ejemplo (cont.)

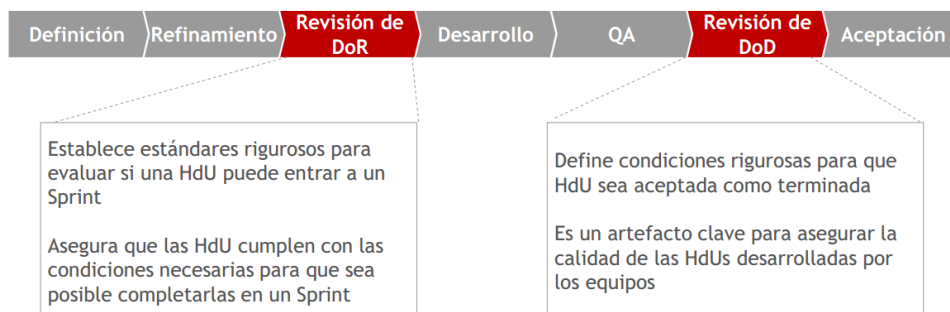
Que solicite salario actual	<b>Escenario 3: Solicitud de Salario</b> <b>DADO</b> que me encuentro en la página de cliente <b>Y</b> se haya registrado al cliente como empleado <b>CUANDO</b> me encuentre ingresando el salario <b>ENTONCES</b> verifique que el valor sea entero y positivo <b>Y</b> menor a \$100.000.000
Que solicite fecha de ingreso a la empresa	<b>Escenario 4: Solicitud de fecha de ingreso</b> <b>DADO</b> que me encuentro en la página de cliente <b>Y</b> se haya registrado al cliente como empleado <b>CUANDO</b> me encuentre ingresando la fecha de ingreso a la empresa <b>ENTONCES</b> la fecha sea superior a 50 años antes a partir de hoy <b>Y</b> la fecha sea inferior a hoy
Que pida tres comprobantes de pago	<b>Escenario 5: Solicitud de 3</b> <b>DADO</b> que me encuentro en la página de cliente <b>Y</b> se haya registrado al cliente como empleado <b>CUANDO</b> me encuentre ingresando los comprobantes de pago <b>ENTONCES</b> solicite tres comprobantes de pago en formato imagen.

11

11

## DoR y DoD

- DoR y DoD son muy importantes en el desarrollo de un producto de software



12

12

## DoR (Definition of Ready)



### Contenido

Debe ser un checklist con estándares claramente definidos y acordados que se deben cumplir



### Responsabilidades

Todo el equipo es responsable por su construcción  
El SM facilita su definición y actualización



### Frecuencia

Al inicio de un equipo recién constituido  
Revisado al fin de cada release



### ¿Cómo se utiliza?

Durante el refinamiento del backlog se utiliza para identificar historias "listas para el sprint".  
Durante el sprint planning se utiliza para aceptar el ingreso al sprint

13

13

## Criterios INVEST

- Se utiliza el criterio INVEST para construir el DoR
- Las HdU deben cumplir con los criterios INVEST

<b>I</b>	<b><u>I</u>ndependiente</b>	Todas las dependencias están claras y resueltas
<b>N</b>	<b><u>N</u>egociable</b>	Es posible negociar sobre la misma tanto en contenido como en prioridad
<b>V</b>	<b><u>V</u>aliosa</b>	Tiene asociada una hipótesis de entrega de valor al negocio, es medible y aporta al objetivo del release
<b>E</b>	<b><u>E</u>stimada</b>	El equipo entiende su complejidad a nivel de tareas y estimó el tamaño de la misma
<b>S</b>	<b><u>S</u>mall</b>	Es lo suficientemente pequeña para ser desarrollada en una fracción del sprint por un desarrollador (i.e. 2 días)
<b>T</b>	<b><u>T</u>esteable</b>	Sus criterios de aceptación son claros y suficientemente exhaustivos. El test es automatizable

14

14

## Ejemplo de DoR

- Sencillo, pero poco exhaustivo

Definition of Ready	
✓	Dependencias claras y definidas (puntos de contacto)
✓	Historias refinadas y criterios de aceptación definidos (2 revisiones)
✓	Complejidad estimada
✓	Historias escritas en JIRA
✓	Historias con documentación adjunta

15

15

## Ejemplo de DoR

- Muy completo

Agrupación	Criterio para que US o HdU ingrese al Sprint Backlog
Valor	<ul style="list-style-type: none"> <li>• Da valor al cliente final (Las historias técnicas evitarlas o justificarlas muy bien)</li> <li>• Está alineada a los objetivos del negocio</li> <li>• Tiene al menos 1 KPI de negocio asociada</li> </ul>
UX/UI	<ul style="list-style-type: none"> <li>• Mockup definido y aceptado</li> <li>• La Experiencia debe ser la mejor posible acorde a los clientes y tendencias</li> </ul>
Agilidad	<ul style="list-style-type: none"> <li>• Es estimada en complejidad</li> <li>• Los criterios de aceptación están definidos, claros y entendidos por todos los miembros del equipo</li> <li>• Pequeña - Que puedan ser completadas máximo en 3 días y se recomienda que tengan entre 3 y 7 criterios de aceptación</li> <li>• Puede ser mostrada por el equipo al cliente y en la Review</li> <li>• Independiente para construir (Lo más independiente posible)</li> <li>• Negociable - Pueda subir o bajar la prioridad según necesidad</li> <li>• Puede ser incluida en el sprint (depende de la velocidad y el tamaño de las otras US)</li> <li>• Equipo tiene las capacidades mínimas para completarla</li> <li>• Cuenta con dependencias claras, críticas están resueltas (Opción usar diagrama de flujo)</li> </ul>
QA	<ul style="list-style-type: none"> <li>• Sus criterios de aceptación pueden ser probados</li> <li>• Sus criterios de aceptación pueden ser automatizables (&gt;75%)</li> </ul>
Herramienta	<ul style="list-style-type: none"> <li>• Está escrita en JIRA con sus adjuntos</li> <li>• Tenemos instalado y con acceso a las aplicaciones necesarias</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• Tiene un diagrama de flujo básico asociado</li> </ul>

15

16



## DoD (Definition of Done)



### Contenido

Debe ser un checklist con estándares claramente definidos y acordados que se deben cumplir



### Responsabilidades

Todo el equipo es responsable por su construcción  
El SM facilita su definición y actualización



### Frecuencia

Al inicio de un equipo recién constituido  
Revisado al fin de cada reléase cuando sea necesario



### ¿Cómo se utiliza?

Al finalizar cada historia el desarrollador debe revisar el DoD  
El PO debe aceptar la misma solo si cumple el DoD. En caso de no cumplirlo la historia se considera no terminada

17

17

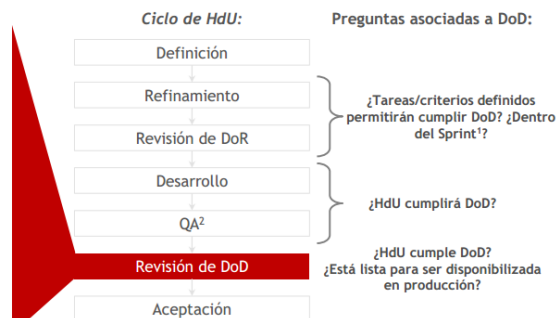
## DoD (Definition of Done)

- El DoD es un artefacto clave para asegurar la calidad de la UdUs desarrolladas por los equipos.

DoD establece estándares rigurosos para considerar las HdU como terminadas

Valor de cara al cliente	<input checked="" type="checkbox"/> Deuda técnica de historias de usuario identificadas <input checked="" type="checkbox"/> Entregables validados por otro equipo o stakeholder
Calidad del código	<input checked="" type="checkbox"/> Todos los test automatizados y manuales (ej. unit tests) <input checked="" type="checkbox"/> Todo el código fue revisado por al menos 3 desarrolladores <input checked="" type="checkbox"/> Funcionalidades inactivas o no implementadas desactivadas <input checked="" type="checkbox"/> Código revisado y analizado por otro desarrollador <input checked="" type="checkbox"/> Jenkins en verde; alarmas implementadas para notificar
Testing	<input checked="" type="checkbox"/> Todas las funcionalidades nuevas cuentan con tests <input checked="" type="checkbox"/> Reportes de stress y carga del sistema ejecutados <input checked="" type="checkbox"/> Branch coverage: >85% para todo el código y 100% para los nuevos
Continuous integration	<input checked="" type="checkbox"/> Código reiniciado y reiniciado (de ser necesario) <input checked="" type="checkbox"/> Integración con clientes y APIs
Compliance	<input checked="" type="checkbox"/> Auditoría (SOX) pasada <input checked="" type="checkbox"/> Check-list de seguridad realizado y pasado
Mobile	<input checked="" type="checkbox"/> Check-list de estándares de la compañía realizada
UX	<input checked="" type="checkbox"/> Todo el código de cara al cliente desarrollado de acuerdo a los estándares
Documentación	<input checked="" type="checkbox"/> Check-list de estándares de la compañía realizada <input checked="" type="checkbox"/> Documentación validada por otra persona antes de su publicación <input checked="" type="checkbox"/> Manual de instalación y configuración para todos los entornos <input checked="" type="checkbox"/> Tickets de JIRA de manuales de instalación en sistema <input checked="" type="checkbox"/> Actualización de los diagramas de los servicios o de la arquitectura <input checked="" type="checkbox"/> Actualización de release notes

DoD se revisa antes que PO acepte HdU, pero se considera desde su desarrollo

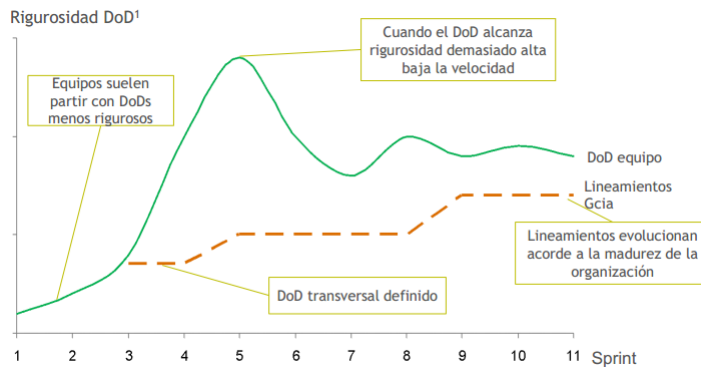


18

18

## DoD (Definition of Done)

- DoDs irán evolucionando para buscar un balance entre calidad y velocidad de los equipos



1 Medida ilustrativa

19

19

## Ejemplo de DoD

### DoD transversal con oportunidades...

Requerimiento (HdU...)	
Valor	<ul style="list-style-type: none"> <li>Permite medir cuantitativamente la hipótesis de valor generado (Ej: Implementando Google Analytics)</li> </ul>
UX	<ul style="list-style-type: none"> <li>Implementa el mockup que fue diseñado; de haber desviaciones, son aprobadas por el PO</li> <li>Cumple con el análisis heurístico de UX</li> </ul>
Calidad del código	<ul style="list-style-type: none"> <li>No genera deuda técnica; de hacerlo, el PO aprueba y documenta en el backlog</li> <li>No tiene errores al correr Checkstyle</li> <li>Implementa cabalmente Normativa de Log</li> <li>Certificada en ambientes pre-productivos</li> </ul>
Arquitectura	<ul style="list-style-type: none"> <li>Debe ser validada por arquitectura en los casos que modifique casos sensibles o el bus de datos</li> </ul>
QA	<ul style="list-style-type: none"> <li>Pasa el 100% de los criterios de aceptación</li> </ul>
Seguridad	<ul style="list-style-type: none"> <li>Pasa requerimientos obligatorios de Normativa de Seguridad</li> </ul>
Inventario/contabilidad	<ul style="list-style-type: none"> <li>Implementa correctamente las transacciones a nivel de inventario y contabilidad</li> </ul>
Monitoreo	<ul style="list-style-type: none"> <li>Define alertas en Mallas y monitoreo de servicios</li> </ul>
Documentación	<ul style="list-style-type: none"> <li>Está marcada como completada en JIRA y en Tablero Scrum</li> <li>HdU aprobada por PO</li> <li>Integrado en repositorio</li> </ul>

20

20

## Producto mínimo viable (MVP)

- “Es un producto construido desde el conjunto de características más pequeñas que entregan valor al cliente, para obtener validación de esas cosas que hacen sentido”.
- Un MVP no consiste en hacer el primer producto malo, sino en centrarse en lo que es realmente importante y construir a partir de eso.

21

21

## MVP



22

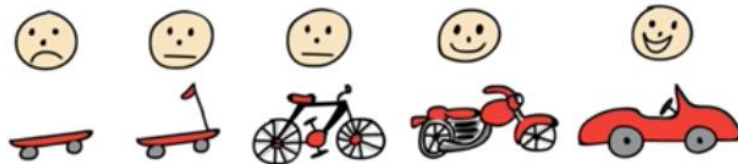
22

# MVP, Modelo Spotify

Not like this...



Like this!



23