

Requisitos

Dos tipos de requisitos

- ▶ Funcionales
 - ▶ qué es lo que el producto o sistema debe ser capaz de hacer (features)
- ▶ No Funcionales
 - ▶ restricciones adicionales mas orientadas al cómo que al qué
 - ▶ performance, usabilidad, mantenibilidad, plataforma

El Antiguo Documento de Requisitos

- 1. Introduction**
 - 1.1 Purpose
 - 1.2 Document conventions
 - 1.3 Intended audience
 - 1.4 Additional information
 - 1.5 Contact information/SRS team members
 - 1.6 References
- 2. Overall Description**
 - 2.1 Product perspective
 - 2.2 Product functions
 - 2.3 User classes and characteristics
 - 2.4 Operating environment
 - 2.5 User environment
 - 2.6 Design/implementation constraints
 - 2.7 Assumptions and dependencies
- 3. External Interface Requirements**
 - 3.1 User interfaces
 - 3.2 Hardware interfaces
 - 3.3 Software interfaces
 - 3.4 Communication protocols and interfaces
- 4. System Features**
 - 4.1 System feature A
 - 4.1.1 Description and priority
 - 4.1.2 Action/result
 - 4.1.3 Functional requirements
 - 4.2 System feature B



- 5. Other Nonfunctional Requirements**
 - 5.1 Performance requirements
 - 5.2 Safety requirements
 - 5.3 Security requirements
 - 5.4 Software quality attributes
 - 5.5 Project documentation
 - 5.6 User documentation
- 6. Other Requirements**
 - Appendix A: Terminology/Glossary/Definitions list
 - Appendix B: To be determined

El problema de las especificaciones

Customer: Hello, I'd like to order a cake.

Employee: Sure, what would you like written on it?

Customer: Could you write "So long, Alicia" in purple?

Employee: Sure.

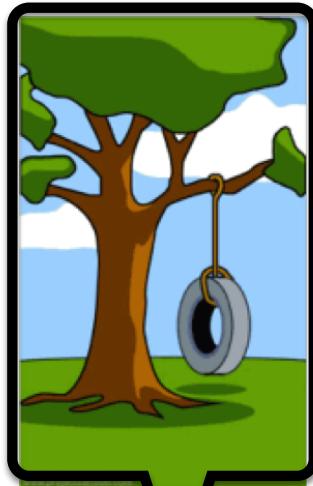
Customer: And put stars around it?

Employee: No problem. I've written this up, and will hand it to my cake decorator right away. We'll have it for you in the morning.



Levantamiento de Requisitos Funcionales

- ▶ Idea - para cada tipo de usuario capturar la forma en que se espera que utilice el producto o sistema
- ▶ Diseño centrado en el usuario
- ▶ Hay dos metodologías ampliamente usadas
 - ▶ Casos de Uso (Use Cases)
 - ▶ Relatos de Usuario (User Stories)



Relatos de Usuario

- ▶ documentos compartidos NO IMPLICA entendimiento compartido
- ▶ el objetivo de los relatos de usuario es el entendimiento compartido
- ▶ usar todos los medios para recordar la conversación

Un relato de usuario es como una foto de vacaciones



Describe lo que ves en la foto

La foto tiene asociado un montón de detalles adicionales que son invisibles para quien no participó

Lecciones Importantes

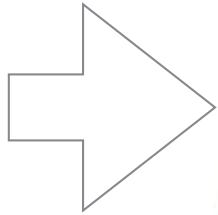
- ▶ los relatos de usuario NO SON un nuevo formato de requerimientos escritos sino un mecanismo para adquirir un entendimiento compartido
- ▶ los relatos representan discusiones sobre solución de problemas de nuestros usuarios y clientes y que llevarán a acuerdos sobre qué construir
- ▶ el objetivo NO ES construir más software más rápido sino **maximizar el outcome** y el impacto

Maximizar Outcome

- ▶ Siempre habrá más que construir que el tiempo y los recursos disponibles
- ▶ Es un error intentar maximizar el output (más features, producto más completo)
- ▶ Debemos apuntar a maximizar el outcome y el impacto ojalá minimizando el output
- ▶ Implica escuchar cuidadosamente a la gente cuyo problema tratamos de resolver

El propósito no es el software !!

- ▶ el software es el *output* del proceso
- ▶ lo que nos interesa realmente es el *outcome*
- ▶ el outcome es el resultado para los usuarios
 - ▶ podrán lograr sus metas con mayor facilidad ?
 - ▶ hemos mejorado en algo sus vidas ?
 - ▶ hay un impacto ? en qué usuarios ?
- ▶ un buen relato no es solo acerca de qué sino de quien y por qué

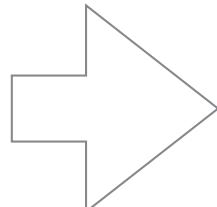


Faltan aún
muchas cosas
que habíamos
conversado

Jefe de Proyecto

Es posible dejar a la gente correcta muy contenta
incluso construyendo una pequeña
fracción de lo que estaba inicialmente planeado

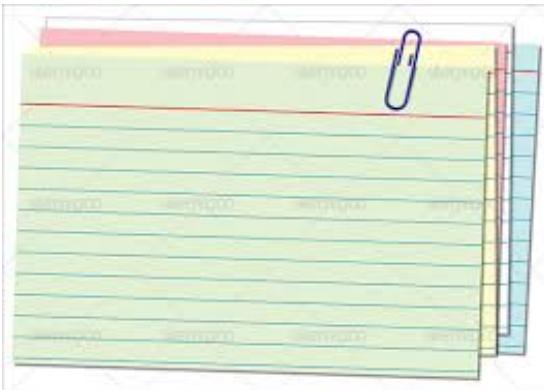
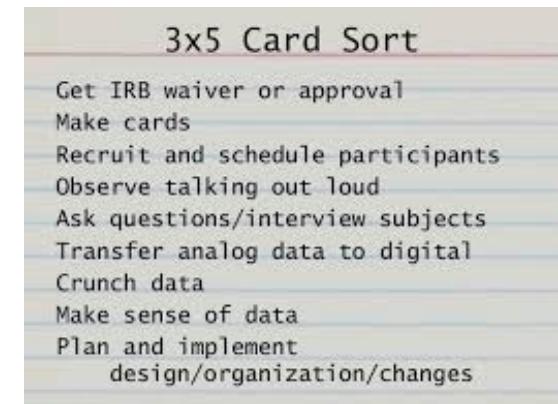
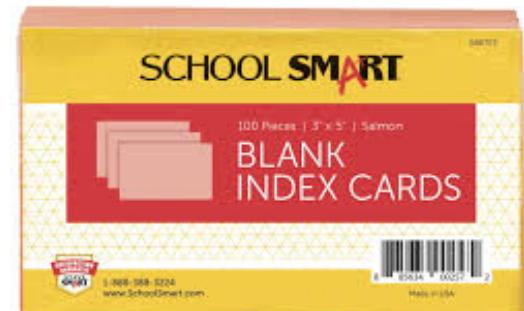
Me van a echar
Falta aún bastante
por hacer



Espectacular,
esto nos
será de gran
ayuda

User Stories (Relatos de Usuario)

- ▶ Muy usada en procesos ágiles (Scrum)
- ▶ Idea es usar una “conversación” para capturar los requerimientos de usuario
- ▶ Puede ser interpretado como una promesa de conversación con el usuario
- ▶ Inicialmente se usaron tarjetas de 3x5 pulgadas (muy populares en USA)



Fundamentos tras la idea

- ▶ requerimientos a través de colaboración
- ▶ requerimientos escritos y pasados en un documento tienen todas las decisiones tomadas
- ▶ con tiempos de entrega cortos es imposible especificar cada detalle de antemano
- ▶ responsabilidad de escribir correctamente los requerimientos se distribuye

El Relato de Usuario como Promesa de Conversación

- ▶ Las tarjetas son solo un punto de partida en el levantamiento de lo que se requiere
- ▶ User Story constituye una promesa de llevar a cabo una conversación (pueden ser varias) con los interesados
- ▶ resultados de la conversación podrían producir algunos documentos adicionales (detalles de la interfaz)

Idea tras los relatos de Usuario

- ▶ el problema de requisitos es realmente un problema de comunicación ...
- ▶ ... entre quienes van a utilizar el software y quienes lo tienen que construir

La Analogía del Médico

Lo que yo necesito
es una operación de
vesícula



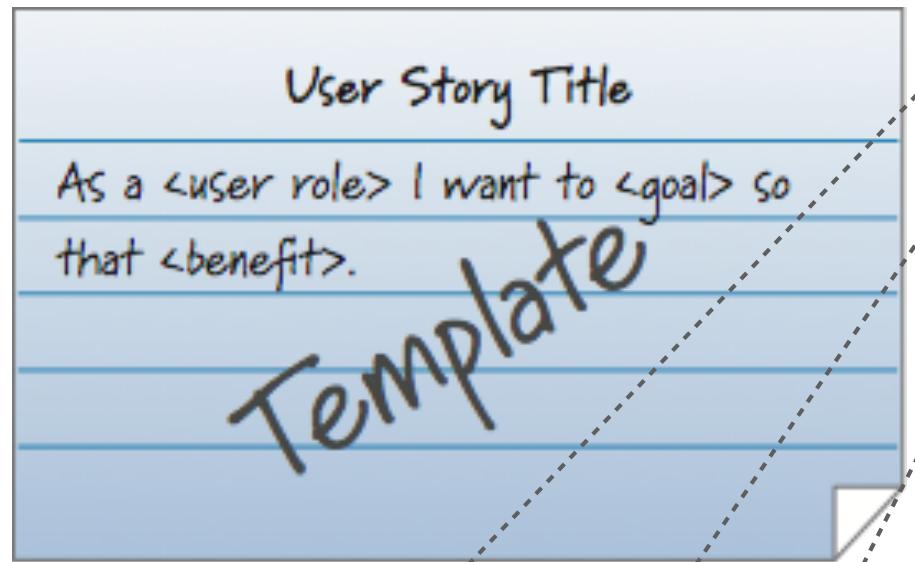
Muy bien, anotaré eso.
¿qué mas necesita?

Ahá.
¿Le duele en forma
permanente?
¿Le duele al levantar el
brazo?
OK
Necesitaremos una
radiografía y
de ahí veremos si necesita
cirugía

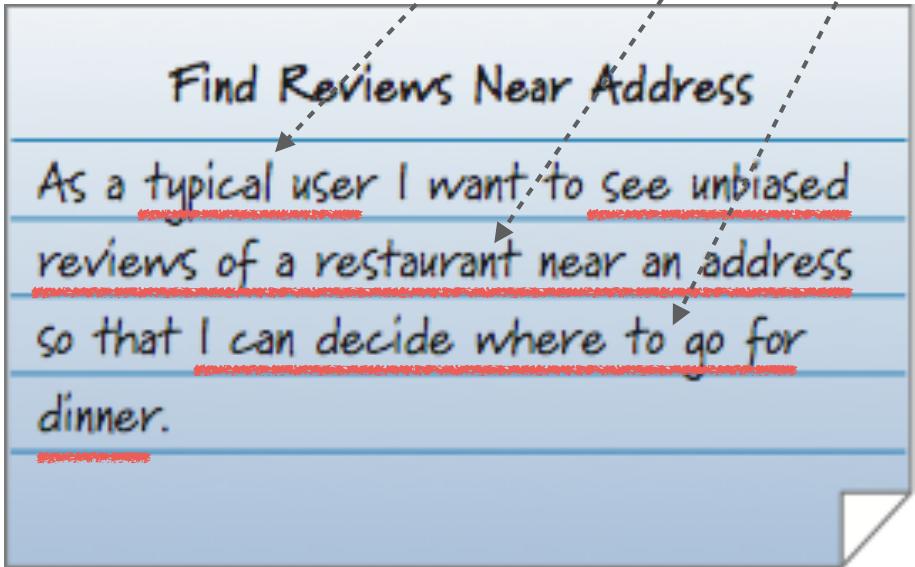


Tengo un dolor en el hombro

Estructura de un Relato

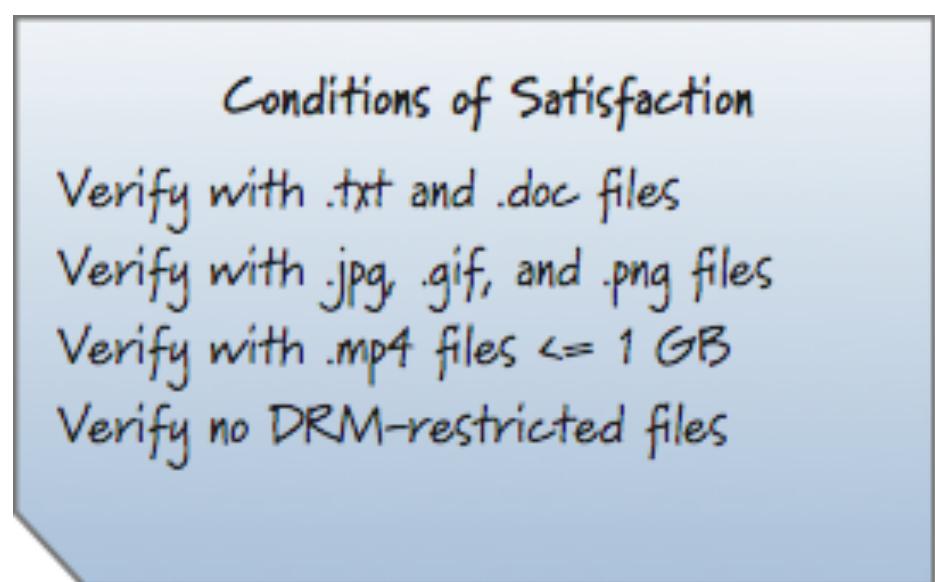
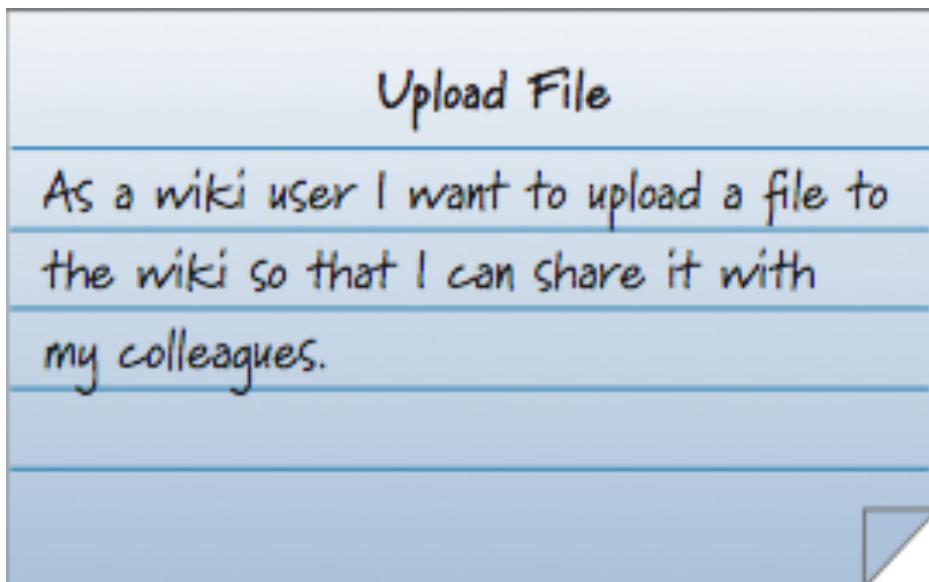


- ▶ user role - categoría de usuario
- ▶ goal - meta de usuario, lo que quiere conseguir
- ▶ benefit - por qué es importante



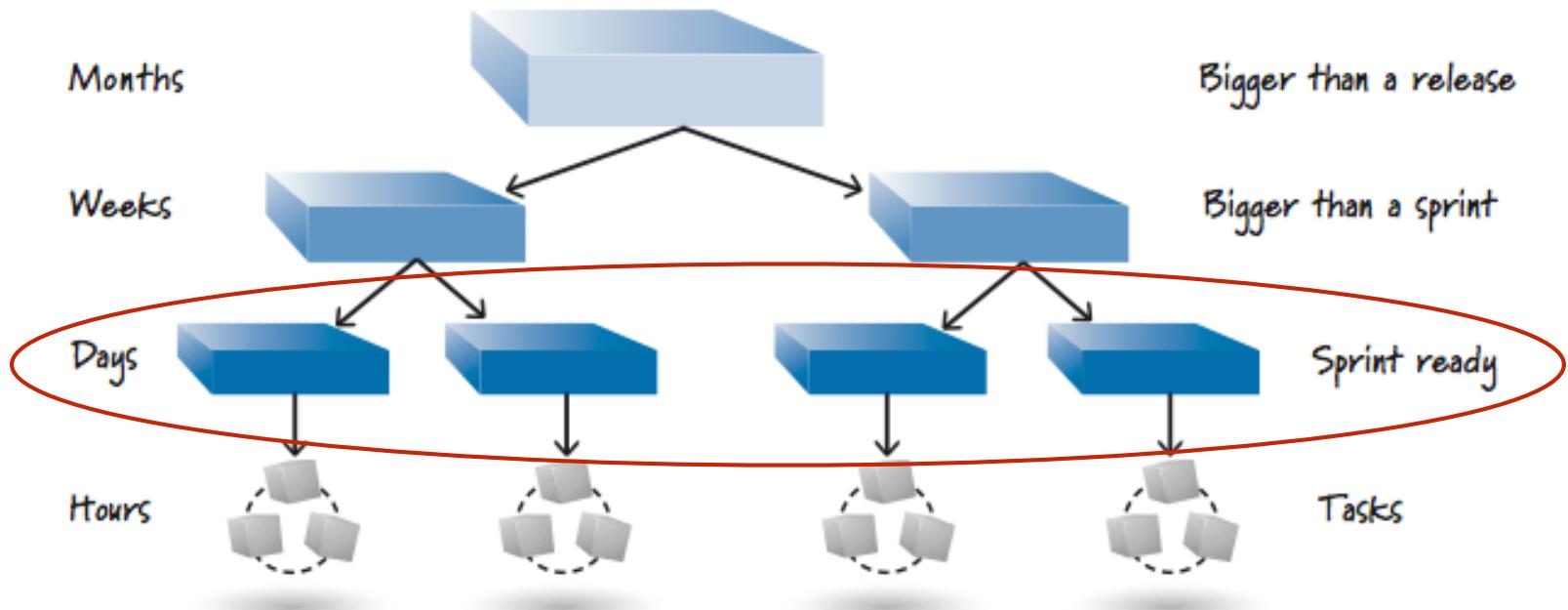
Confirmación

- ▶ En el reverso de la tarjeta se ponen las condiciones de satisfacción (SC)
- ▶ SC pueden ser expresados como un test de aceptación



Nivel de Detalle

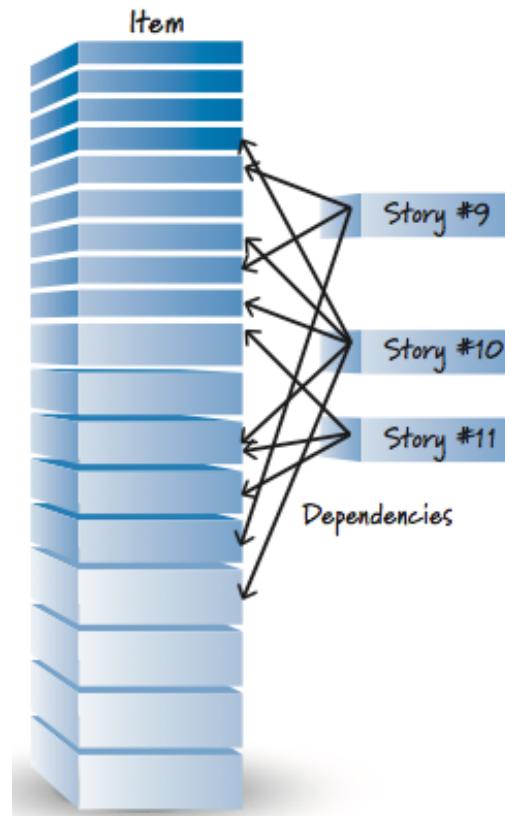
- ▶ pueden usarse varios niveles de abstracción



- ▶ nos interesan relatos a nivel de sprint (iteración)

Ojalá sean independientes

- ▶ Se hace muy difícil priorizar si hay muchas dependencias entre las tareas
- ▶ tratar de escribir las tareas de modo que se minimicen las dependencias



Deben tener un valor para el cliente

- ▶ si un relato no es de valor para nadie no se debe hacer
- ▶ cuidado con cosas que son de valor para desarrolladores (valor técnico)
 - ▶ migrar código a última versión de Rails
- ▶ en esos casos es posible a veces reformularlo de forma que tenga un claro valor para el cliente
 - ▶ migrar código para mitigar riesgo

Estimable

- ▶ Si el equipo no es capaz de estimar el esfuerzo para desarrollar un US puede ser que
 - ▶ US sea muy grande
 - ▶ US sea ambiguo

¿Como escribir las historias ?

- ▶ incorporar a los usuarios al equipo en esta etapa
- ▶ workshop - brainstorming con participación de usuarios (al menos product owner)
- ▶ puede ser de unas horas hasta unos pocos días
- ▶ la idea es generar un set inicial y no un set completo
- ▶ comenzar identificando roles de las historias para poder escribir “As a <user role>, I want to ...”
- ▶ top-down o bottom-up

Errores Comunes

- ▶ tratar de evitar el genérico "as a user" (ser más específico)
- ▶ relato es muy pequeño
 - ▶ resultados deben guardarse como archivo XML
 - ▶ resultados deben guardarse como HTML
- ▶ relato con interdependencias (difícil planeación)
- ▶ Goldplating (haciendo más de lo necesario)
 - ▶ además de tabs se pueden arrastrar
- ▶ Demasiado detalle (por eso se usan tarjetas pequeñas)
- ▶ Detalles de interfaz de usuario demasiado pronto
- ▶ traspasar requisitos formales a relatos de usuario

Ejemplo de Relatos

As a user, I want to
reserve a hotel room.

As a vacation traveler,
I want to see photos of
the hotels.

As a user, I want to
cancel a reservation.

As a frequent flyer, I
want to rebook a past trip
so that I save time
booking trips I take often.

Y los detalles

As a user, I want to
cancel a reservation.

- ▶ se le debe hacer una devolución completa ?
 - ▶ si pagó con tarjeta ?
- ▶ hasta cuando se acepta la cancelación
 - ▶ hay diferencias entre hoteles ?
 - ▶ hay diferencias entre usuarios (viajeros frecuentes) ?
- ▶ se le debe dar una conformación al usuario ?

Condiciones de Satisfacción

- ▶ Pueden ser usadas para agregar detalles al relato
 - ▶ verificar que un miembro premium pueda cancelar el mismo día sin cargo
 - ▶ verificar que miembros no premium deben pagar un 10% si cancelan el mismo día
 - ▶ verificar que el email de conformación se envía
 - ▶ verificar que el hotel sea notificado de la cancelación

A veces el detalle aparece en sub-relatos

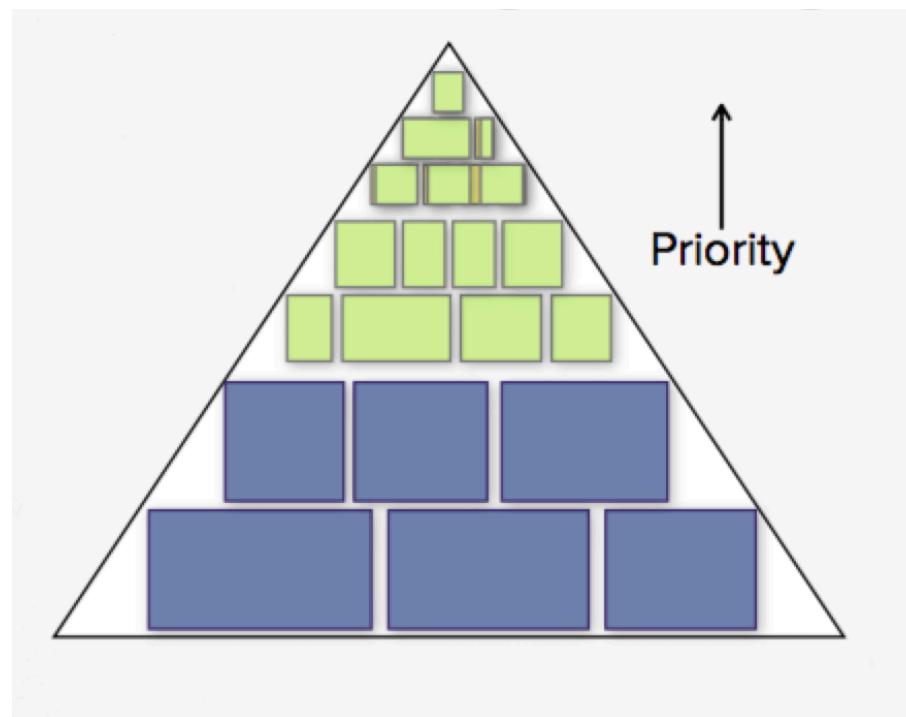
As a user, I can cancel a reservation.

As a premium site member, I can cancel a reservation up to the last minute.

As a non-premium member, I can cancel up to 24 hours in advance.

As a site visitor, I am emailed a confirmation of any cancelled reservation.

El backlog del producto



Epicas, Temas y Relatos

- ▶ La idea es ver los relatos de usuario en un contexto mayor que una sola pequeña parte
- ▶ Layout en una pared, o en un tablero
- ▶ Agrupar en *épicas* y *temas*
- ▶ Imaginar al producto como si existiese y describir una historia completa
 - ▶ comprar un ticket de avión a San Francisco
 - ▶ tomar mis cursos del semestre 2'2019 (Banner)
- ▶ Las tareas necesarias para llevar a cabo ese relato se ponen de izquierda a derecha y de arriba a abajo en tarjetas

Epicas vs Temas

- ▶ Una épica corresponde a un objetivo mayor que requiere ser separado en varios relatos que deben hacerse en secuencia
- ▶ Por lo general los relatos que son parte de una épica no tienen mucho valor en sí mismos
- ▶ Un tema puede ser cualquier agrupación de relatos (variaciones de un mismo objetivo, etc)
- ▶ Los relatos de un tema tienen valor en forma independiente

Para ponerlo en términos muy sencillos

- ▶ Tema : una colección de relatos relacionados de algún modo
- ▶ Epica : un relato grande que puede ser dividido en varios pasos

Ejemplo

1. Realizar una transferencia bancaria

Epica

1.1.autenticarse como cliente del banco

Relato

1.2.especificar el destino de la transferencia

- ingresar datos de nuevo receptor
- indicar datos de un receptor ya usado antes
- ingresar mensaje asociado

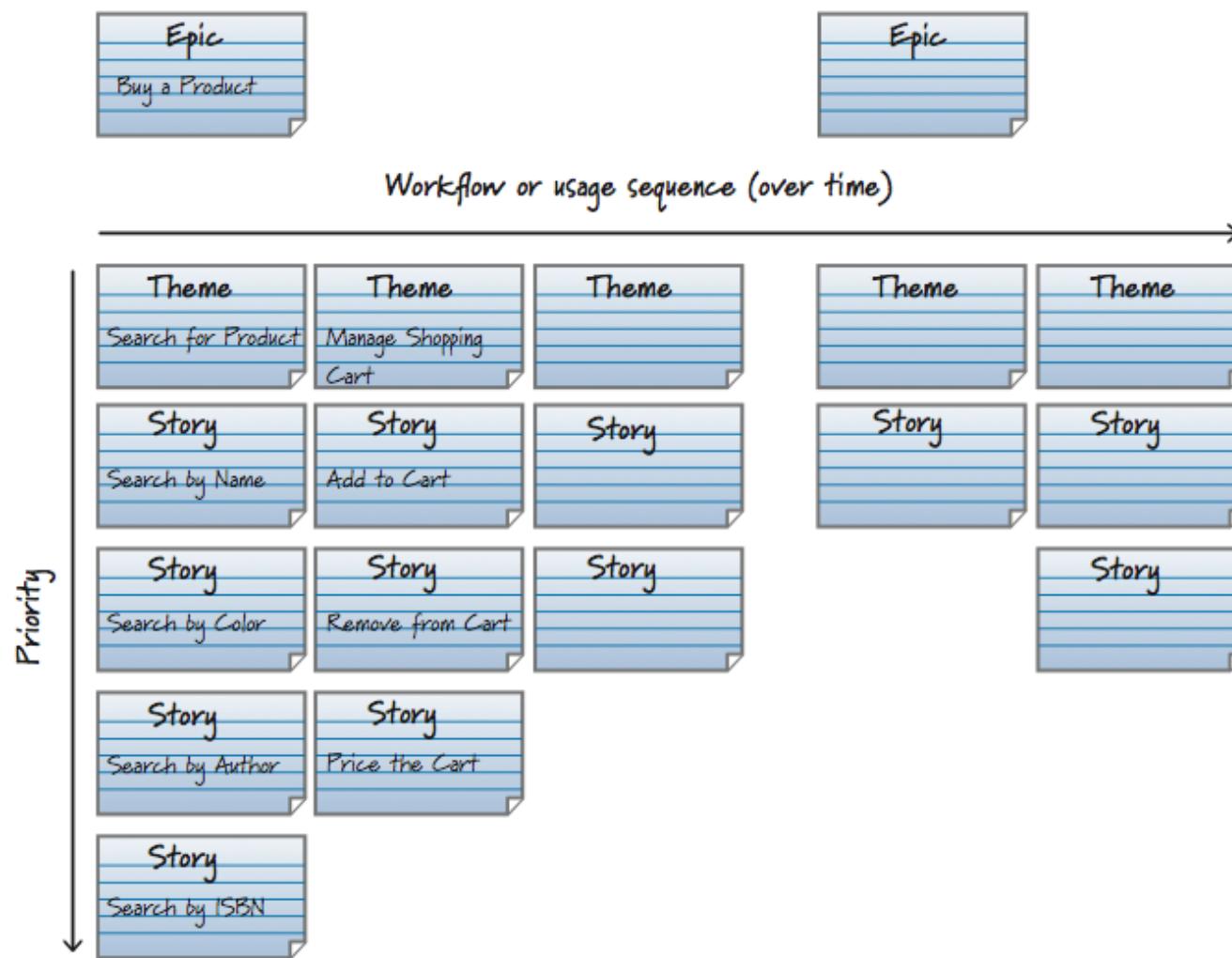
1.3.especificar la cuenta de origen

1.4.realizar la transferencia propiamente tal

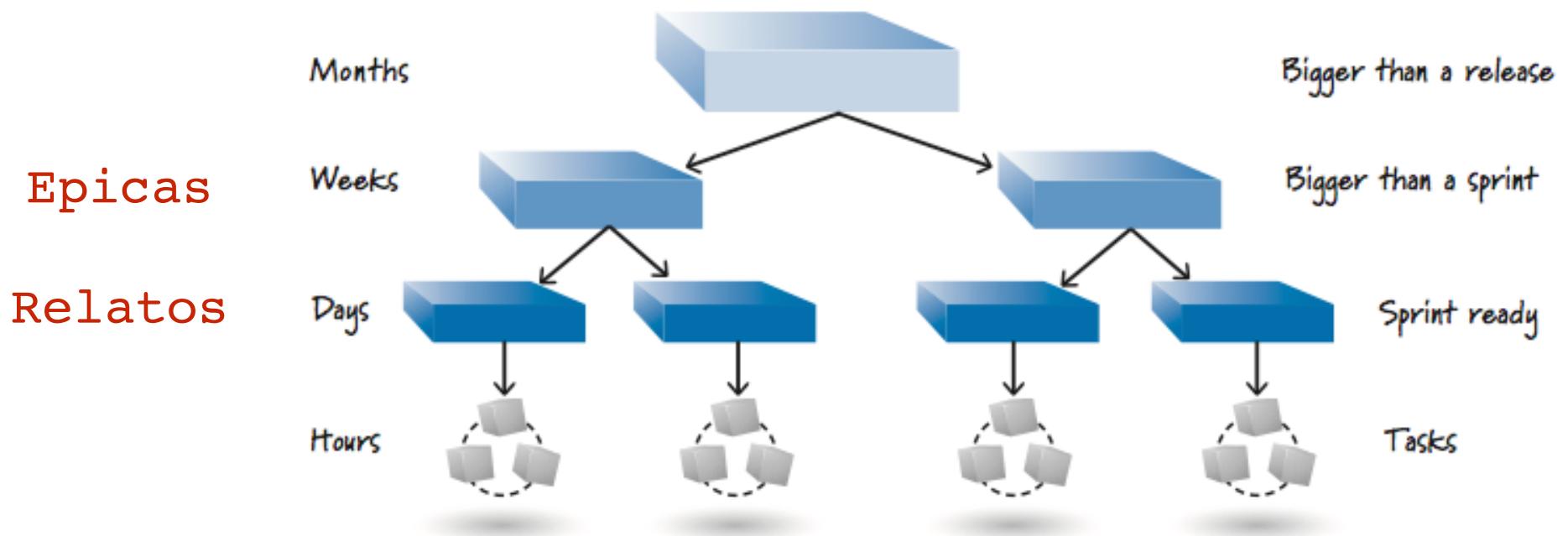
- ingresar pinpass
- ingresar coordenadas

Story Mapping

- ▶ Una épica es un relato mayor compuestos de muchos relatos pequeños
- ▶ Relatos pequeños pueden agruparse en temas

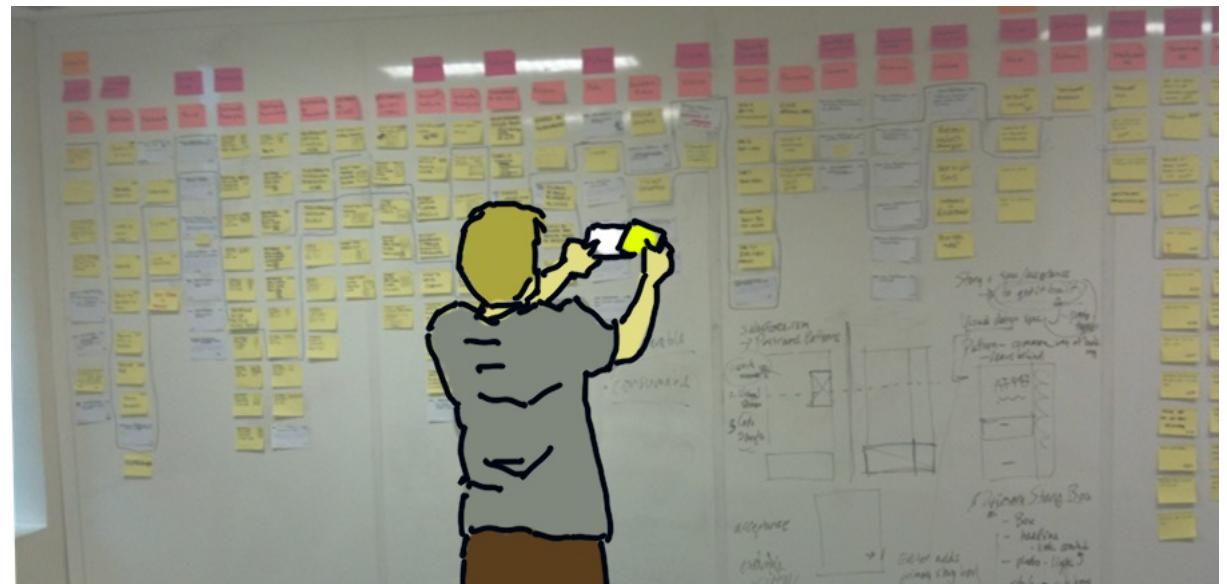
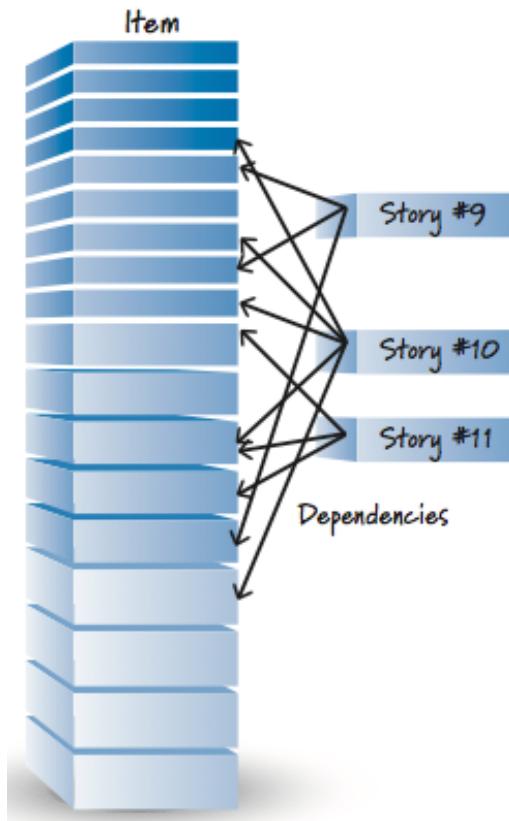


Una épica corresponde a un relato demasiado grande



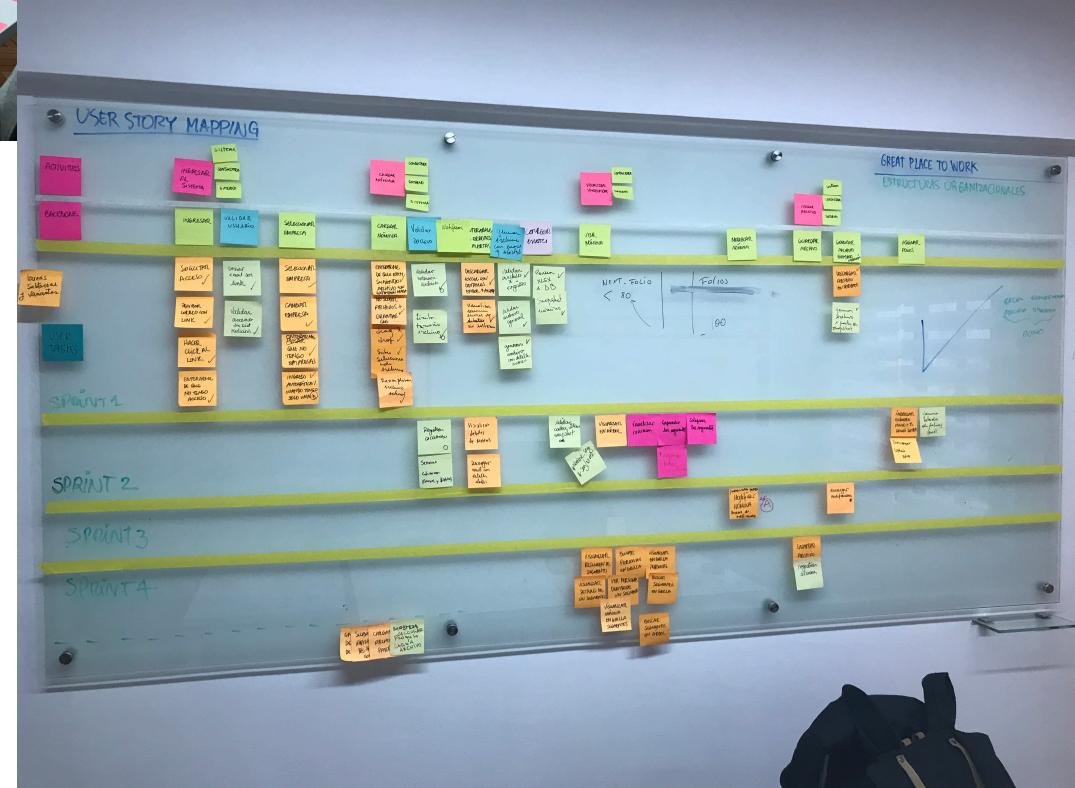
Visualización concreta de lo que hay que hacer

- ▶ Se hace muy difícil priorizar si hay muchas dependencias entre las tareas
- ▶ tratar de escribir las tareas de modo que se minimicen las dependencias





Abril 2018



Ejemplo

- ▶ Visualizar un sistema que sirva de ayuda a los alumnos de ingeniería para navegar a través de las múltiples opciones: majors, minors, títulos, postgrados
- ▶ Escribir épicas
- ▶ Generar relatos de usuario para dos de estas épicas
 - ▶ incluir criterios de prueba en el relato

Posibles Epicas

- ▶ Apoyar al alumno en su decisión sobre el Major
- ▶ Apoyar al alumno en su decisión sobre Minors
- ▶ Apoyar al alumno próximo a completar el primer ciclo en decisión sobre segundo ciclo

Relatos

- ▶ Epica: Apoyar al alumno en su decisión sobre el Major
 - ▶ Obtener información de majors disponibles
 - ▶ Obtener los cursos que incluye un major
 - ▶ Obtener número de alumnos que ingresan anualmente a un major
 - ▶ Dado un mayor obtener comentarios de alumnos de ese major
 - ▶ Dado un major obtener posibles articulaciones con títulos profesionales

Descripción

Obtener información de majors disponibles

Como alumno que aún no ha optado por un major quiero una lista de todos los major disponibles con una breve descripción y el Departamento responsable de darlo para formarme una visión amplia de la oferta disponible

Criterio de Prueba

Al solicitar se devuelve la lista de la totalidad de los majors disponibles con un breve resumen y el Depto responsable

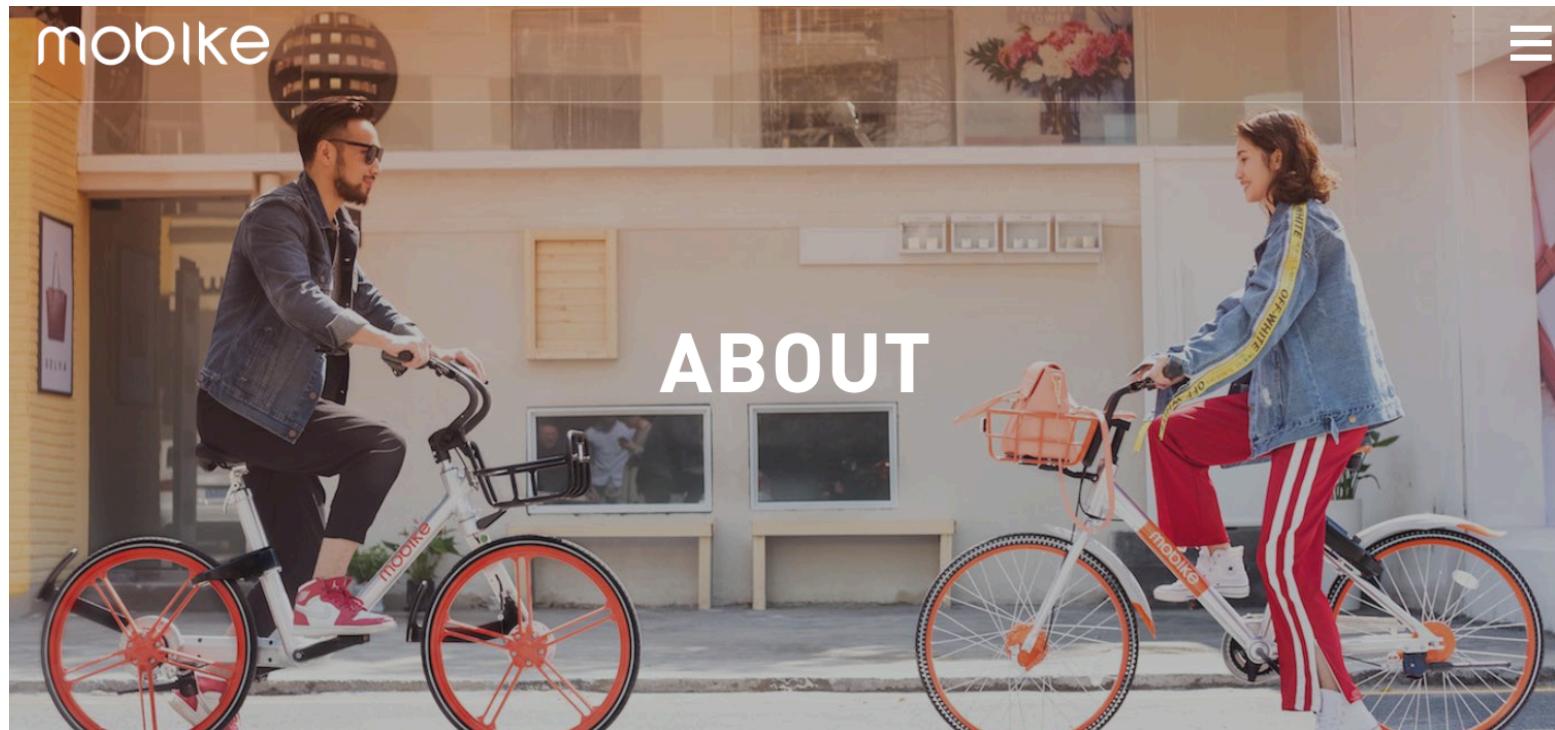
Descripción

Obtener los cursos que incluye un major

Como alumno quiero poder buscar un major para solicitar los cursos que el incluye para ver si el programa de estudios es compatible con mis intereses

Criterio de Prueba

Se ingresa el nombre de un major y se despliega la lista de cursos contenidos en el (los cursos podrían ser links a sus páginas respectivas para mayor información)



ABOUT

What is Mobike?

Imagine a world where you can pick up and leave a bike at your convenience. Enter Mobike: a bike sharing service to fulfil urban short trips - anytime, to any legal parking destination - by combining innovation and today's IoT (Internet of Things) technology. Mobike is green, reduces congestion, and continually strives to improve the quality of city life.

Mobike's Mission

Mobike provides an affordable means of shared transportation for convenient short urban trips, while reducing congestion, and our city's carbon footprint. These combined - Mobike improves the quality of city life.



Título de una épica

- ▶ Una línea que comienza con un verbo y describe la gran meta de alguno de los actores
 - ▶ Obtener una bicicleta para viajar de A a B
 - ▶ Registrarse en la aplicación
 - ▶ Obtener información de ubicaciones de bicicletas para ser reubicadas
 - ▶ Obtener estadísticas de uso de las bicicletas

Epicas vs Temas

- ▶ Una épica es similar a un relato en que hay un objetivo o meta clara
- ▶ Los relatos de una épica constituyen metas parciales que se requieren para la meta mayor
- ▶ Hay una cierta secuencia de los relatos
- ▶ Un tema es una agrupación arbitraria de relatos

Ejemplo de Epica

- ▶ Epica: Obtener una bicicleta para viajar de A a B
- ▶ Relatos asociados
 - ▶ Obtener información de bicicletas cercanas
 - ▶ Autenticarse en la aplicación
 - ▶ Desbloquear una bicicleta

Ejemplo de Tema

- ▶ Tema: Estadísticas para la gestión
- ▶ Relatos Asociados
 - ▶ obtener número de viajes y número de biciletas usadas en una ciudad y por sector
 - ▶ obtener gráficos de crecimiento: número de inscritos, dinero recibido mensualmente
 - ▶ obtener detalle de biciletas reubicadas cada día