

# **Ingeniería de Software I**

## **Segundo Cuatrimestre de 2016**

Clase 3: Métodos “light” para especificar requerimientos

Buenos Aires, 18 de agosto de 2016

# El proceso de los Requerimientos



# Extracción de Información

- ▶ Problemas
  - ▶ La gente no sabe que es lo que sabe
  - ▶ La gente no sabe expresar o explicar lo que sabe
  - ▶ La gente usa terminologías distintas
  - ▶ No todos saben todo
  - ▶ Las personalidades más fuertes dominan a las más débiles
- ▶ Siempre empiecen por estas preguntas
  - ▶ ¿Quiénes son “stakeholders”?
  - ▶ ¿Cuál es el propósito del sistema?
- ▶ Busquen una respuesta corta

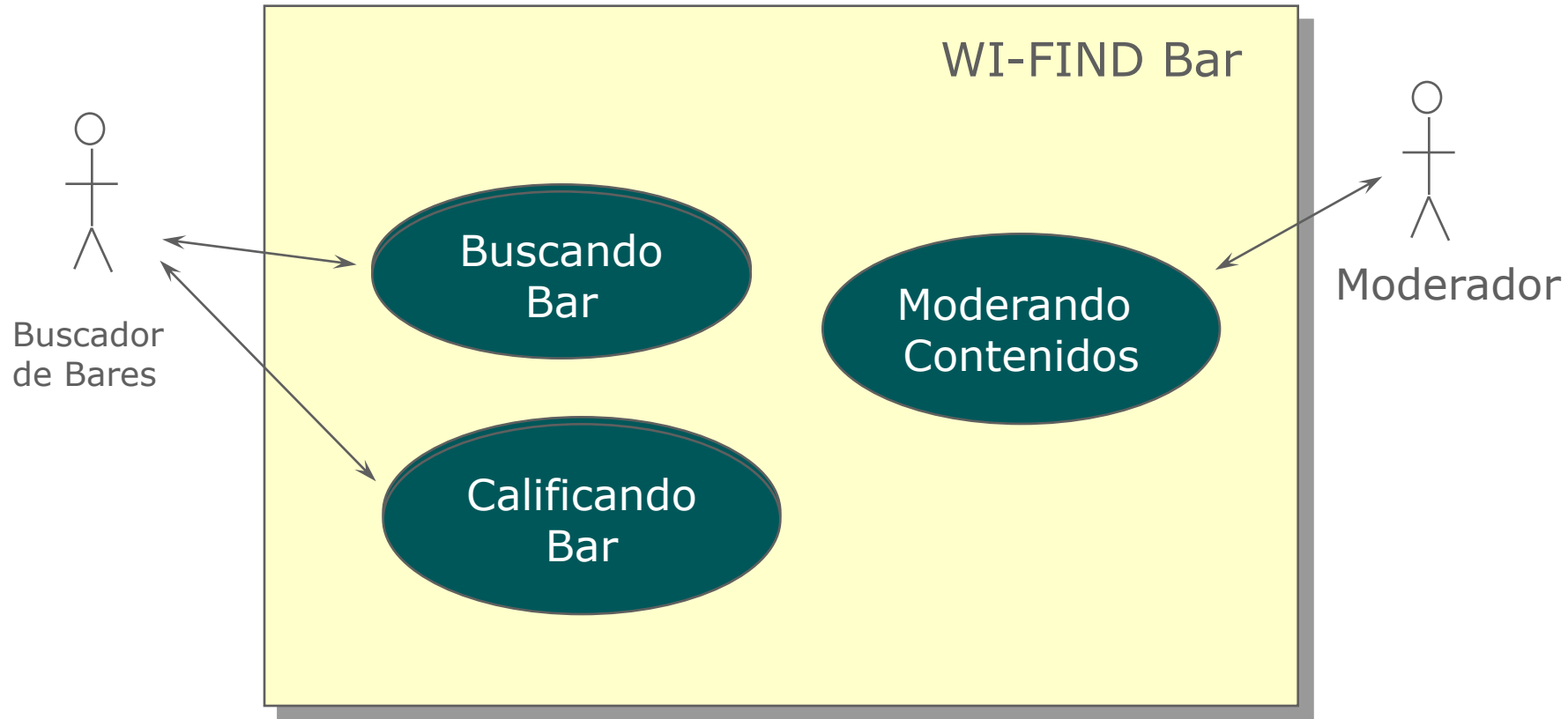
## Modelado con Casos de Uso

- ▶ *Caso de uso*: secuencia de eventos relacionados que implican interacción entre el sistema y algo externo a él
  - ▶ “algo” es una persona u otro sistema
- ▶ Es como un resumen de un “manual de usuario”, con menos detalles de implementación
- ▶ Se documentan en lenguaje natural
- ▶ Propuestos originalmente por Jacobson en 1992 (Object Oriented Software Engineering, a Use Case Driven Approach)

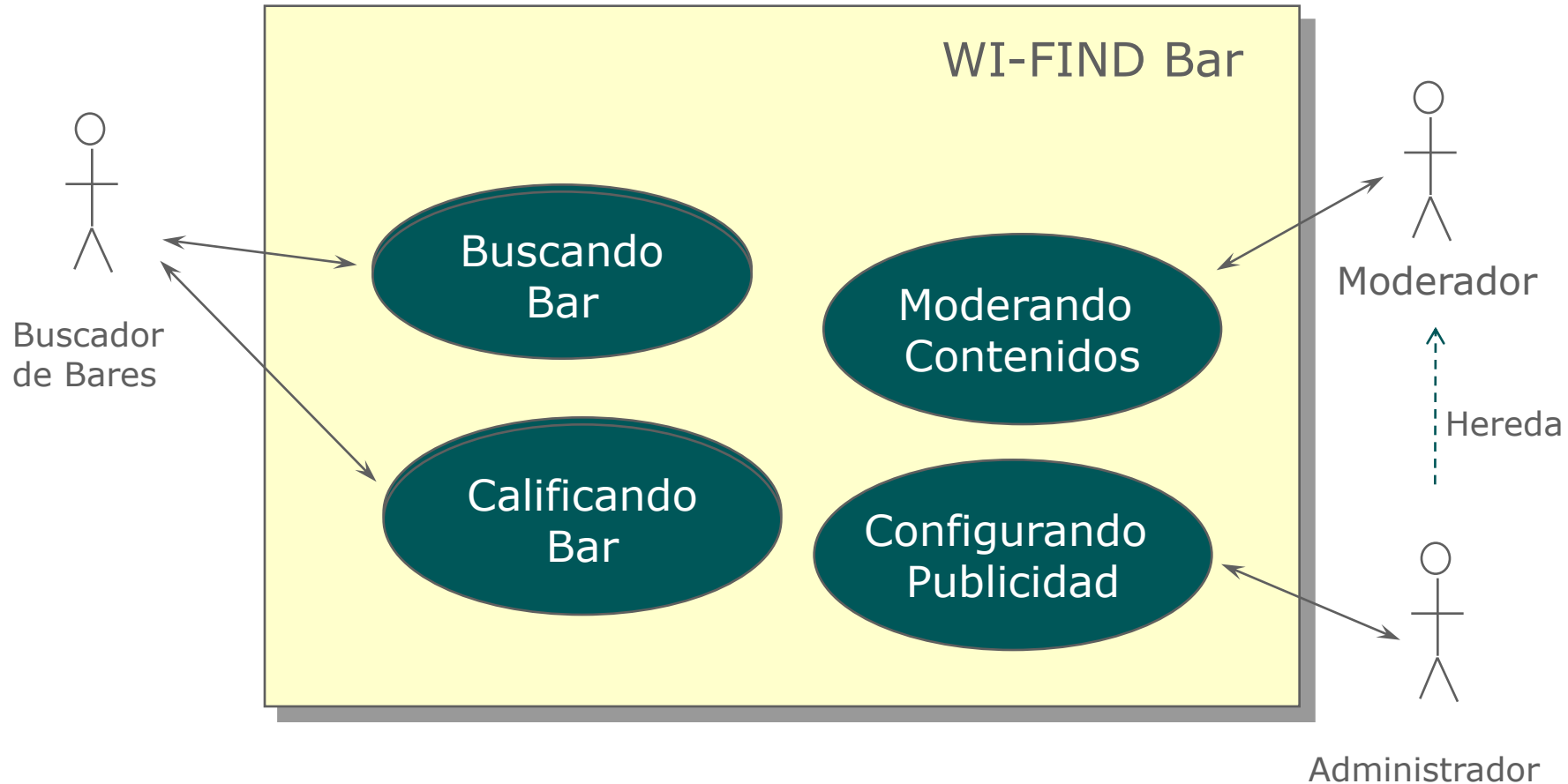
# Actores

- ▶ Todo usuario o sistema que interactúe con nuestro sistema es un actor
- ▶ Un actor es un rol
- ▶ Un usuario es una instancia de un actor. El mismo usuario puede instanciar a distintos actores
- ▶ Los principales actores son fáciles de identificar a partir del propósito del sistema
- ▶ ¿Actores del TP?

# Ejemplo



# Ejemplo herencia



## Cómo identificar Casos de Uso

- Empezamos por casos “triviales”, que surgen del propósito del sistema
- Hacemos las siguientes preguntas:
  - ¿Existen variaciones significativas del caso, en función por ejemplo del actor o del momento en que se ejecute?
  - ¿Existen casos de uso que precedan a este?
  - ¿Existen casos de uso que sucedan a este?
  - ¿Existen casos de uso opuestos a este?
    - Caso opuesto y negación del caso



## Ejemplos

- ▶ Pensemos en el caso de uso “Calificando Bar”
  - ▶ ¿Variaciones según tipo de usuario?
  - ▶ ¿Caso que lo precede?
  - ▶ ¿Caso que lo sucede?
  - ▶ ¿Caso contrario?

## Duración de los casos

- ▶ Una duda común es “¿Cuánto dura un caso de uso”?
- ▶ Respuesta: depende
  - ▶ En una primera etapa se pueden hacer documentar casos más extensos, por ejemplo, todo el proceso de un crédito desde que se lo solicita hasta que se lo cancela
  - ▶ Más adelante, si no quiero casos de uso “spaghetti”, debo modularizarlos

## Modularización de casos de uso

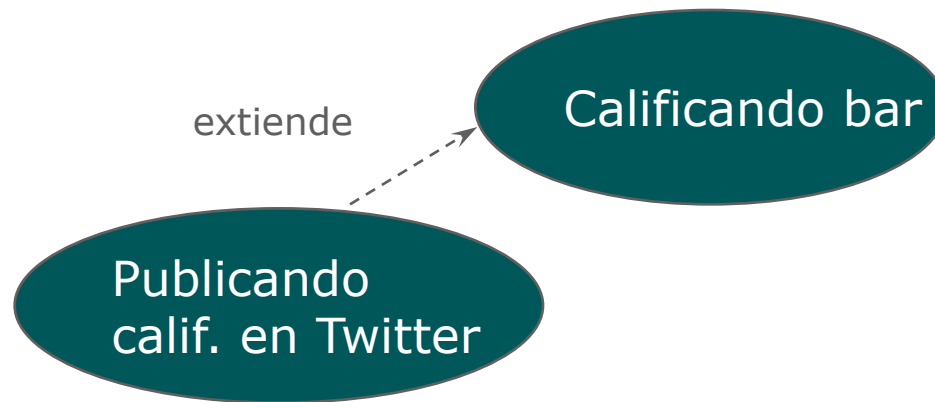
- Alternativas
  - Todo caso de uso tiene un curso normal, “si todo va bien”
  - Las desviaciones del curso normal se llaman alternativas
  - No tienen sentido en forma aislada
  - En general son errores o excepciones
  - Los casos y sus alternativas se documentan en tablas

## Alternativas - Ejemplo

Caso de Uso: Reportando Bar	
Curso Normal	Alternativas
1) El usuario selecciona la opción para reportar un bar	
2) El sistema carga datos básicos sobre el bar, calificando wifi, enchufes, ambiente y comida	
3) El sistema pregunta si confirma la información reportada	3.1) Si el usuario dice que no confirma, vuelve a la pantalla de edición
4) El sistema guarda la información reportada y le informa al usuario que está pendiente de revisión	
5) El usuario contesta si quiere reportar otro bar	5.1) Si contesta que si, vuelve a la opción de reportar un bar
6) Fin del caso	

## Extensiones

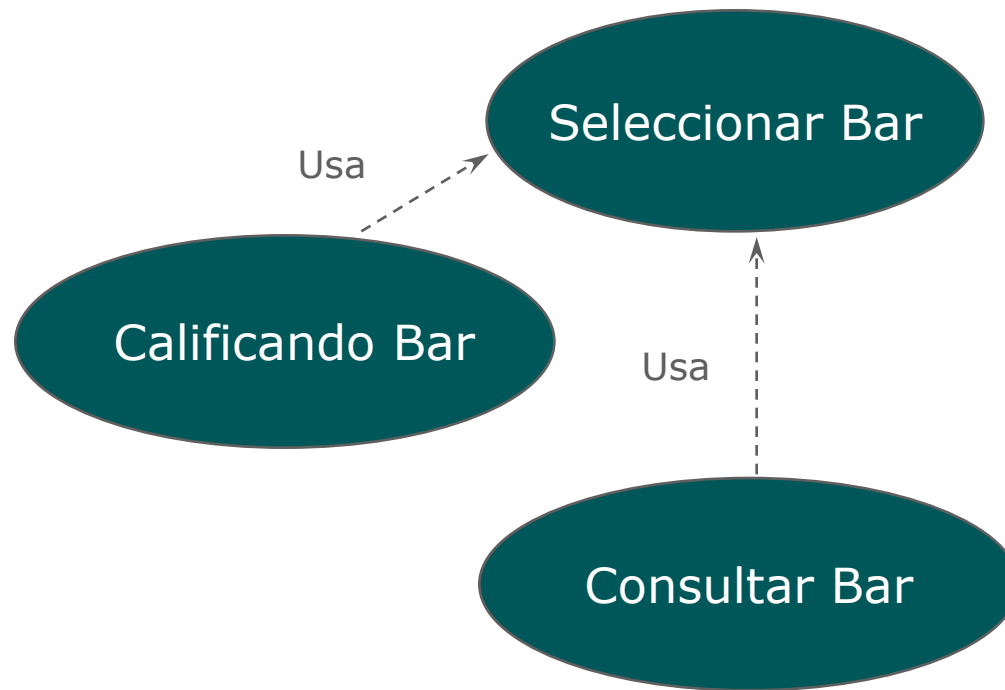
- ▶ Cuando partes completas de un caso de uso sólo ocurren en determinadas situaciones, las llamamos “extensiones”
- ▶ Una extensión es un caso de uso en sí mismo



## Relaciones de uso entre casos

- ▶ Cuando en varios casos de uso detecto funcionalidad en común, podemos crear casos de uso “usados” por otros casos
- ▶ Esos casos “usados” también deben tener sentido en sí mismos

## Relaciones de uso - Ejemplo



Podemos crear “actores abstractos”

## Usa - Extiende - Alternativas

- ▶ El caso que extiende sólo se ejecuta bajo ciertas condiciones
- ▶ El caso que es usado siempre se ejecuta cuando se ejecuta el caso que lo usa
- ▶ Ambos son casos de uso en si mismos
- ▶ Las alternativas son simples desviaciones en el caso normal, que no tienen sentido fuera del contexto de ese caso de uso



## Tipos de Casos de Uso

- ▶ Al principio, conviene definir casos de uso “de trazo grueso”
  - ▶ No tienen en cuenta detalles de implementación
  - ▶ Tienen sólo las alternativas más importantes
- ▶ Una vez validados los casos de trazo grueso, hago los casos de “trazo fino”
  - ▶ Incluyo detalles sobre la implementación (ojo, esto está sujeto a cierto debate)

## Tipos de Trazos – Ejemplo “Calificando Bar”

Evento: Calificando bar	
Trazo Grueso	Trazo Fino
1) El usuario califica los ítems con puntuación	1) El usuario califica los ítems Wifi, Enchufes, Comida, Precio y ambiente en una escala de 1 a 5 estrellas
2) El usuario ingresa texto libre	2) El usuario ingresa (opcionalmente) un comentario en texto libre de hasta 500 caracteres sin espacio. El sistema valida longitud máxima.
3) El usuario puede ingresar fotos	3) El usuario puede hacer un “upload” de hasta 5 fotos de hasta xMb de tamaño....
4) El usuario puede publicar en twitter su reporte	4) Extensión: publicando calificación en twitter

## El Proceso según Rumbaugh

1. Identificar los límites del sistema
2. Clasificar los objetos externos según su rol en la aplicación (identificar actores)
3. Para cada actor, pensar en las principales formas diferentes en las que usa el sistema (identificar casos de uso a nivel de nombre)
4. Pensar algunos escenarios para cada caso
5. Determinar las secuencias de interacción: Evento inicial, precondiciones, finalización normal
6. Escribir una descripción en prosa del caso
7. Considerar todas las excepciones que pueden ocurrir en el desarrollo del caso, y cómo lo afectan
8. Buscar fragmentos comunes en los casos, para identificar extensiones o usos

## **Ejemplos de Casos de Uso “en la vida real”**

## User Stories

- ▶ Son la “forma sugerida” para documentar requerimientos en varios métodos ágiles
- ▶ Las 3C de Ron Jeffries
  - ▶ Card: Se documentan en tarjetas. Se complementan con estimaciones, notas, etc.
  - ▶ Conversation: los detalles surgen de “conversaciones” con los usuarios
  - ▶ Confirmation: Tests de aceptación registran los “detalles”

## Ejemplos para nuestro TP

- ▶ Un usuario puede calificar un bar
- ▶ Un usuario puede buscar bares cercanos
- ▶ Un usuario puede "browsear" bares
- ▶ Un administrador puede configurar publicidad
- ▶ Un usuario puede compartir calificaciones en redes sociales
- ▶ Template sugerido:
  - ▶ Como USUARIO quiero OBJETIVO tal que MOTIVO

## ¿Y los detalles?

- Sub – Stories
- Acceptance (test) cases

## ¿Cómo empiezo?

- Identificando roles de usuario (exactamente lo mismo que “actores”)
- Valores específicos para atributos que caracterizan una población de usuarios y que determinan su interacción con el sistema
  - ¿Ejemplos para nuestro TP?

## Personas

- ▶ En “diseño centrado en el usuario”
- ▶ Concepto incorporado al software por Alan Cooper
- ▶ Representación imaginaria (extensión) de un rol de usuario

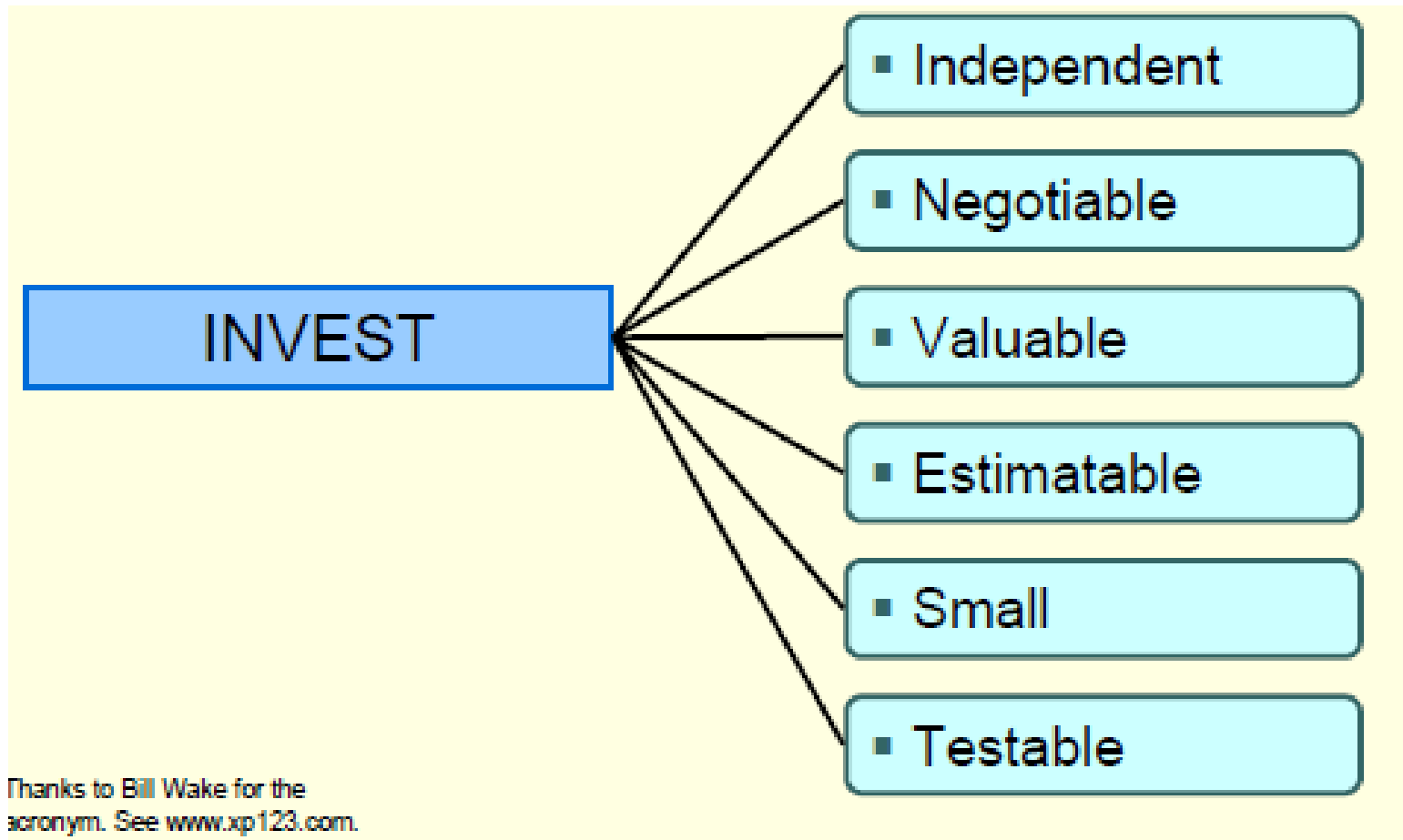


# Eugenio

- ▶ Eugenio tiene 22 años y aunque tiene internet en la casa siempre busca bares donde poder navegar gratis
- ▶ Mira la información sobre velocidad de internet y enchufes con máximo detalle
- ▶ No le importa ni el café ni la comida ni nada parecido en el bar
- ▶ Siempre selecciona los bares en función de distancia, mirando el mapa, y la información de conexión
- ▶ Quiere publicar la información sobre los bares en la máxima cantidad de redes sociales que sea posible

De a poco reemplazamos “El usuario” por “personas” (por nombre o característica principal)

## El Modelo INVEST



## User Story con Criterios de Aceptación

#1 - As a buyer, I want to place a book in a shopping cart so I can choose the books that I will buy

Acceptance criteria:

1. User can select a book from the catalog
2. User can search the catalog by author or title.
3. If the book is already in the shopping cart, its quantity is increased.
4. If the book is not already in the shopping cart, it is added with a quantity of one.
5. After the operation, the shopping cart is displayed.

Effort: 5

Value: 7