



Ingeniería Software

por María Paula Herrero

13

DISCIPLINA
REQUISITOS



CONTENIDO

¿Qué es un requisito?

¿Qué es un sistema?

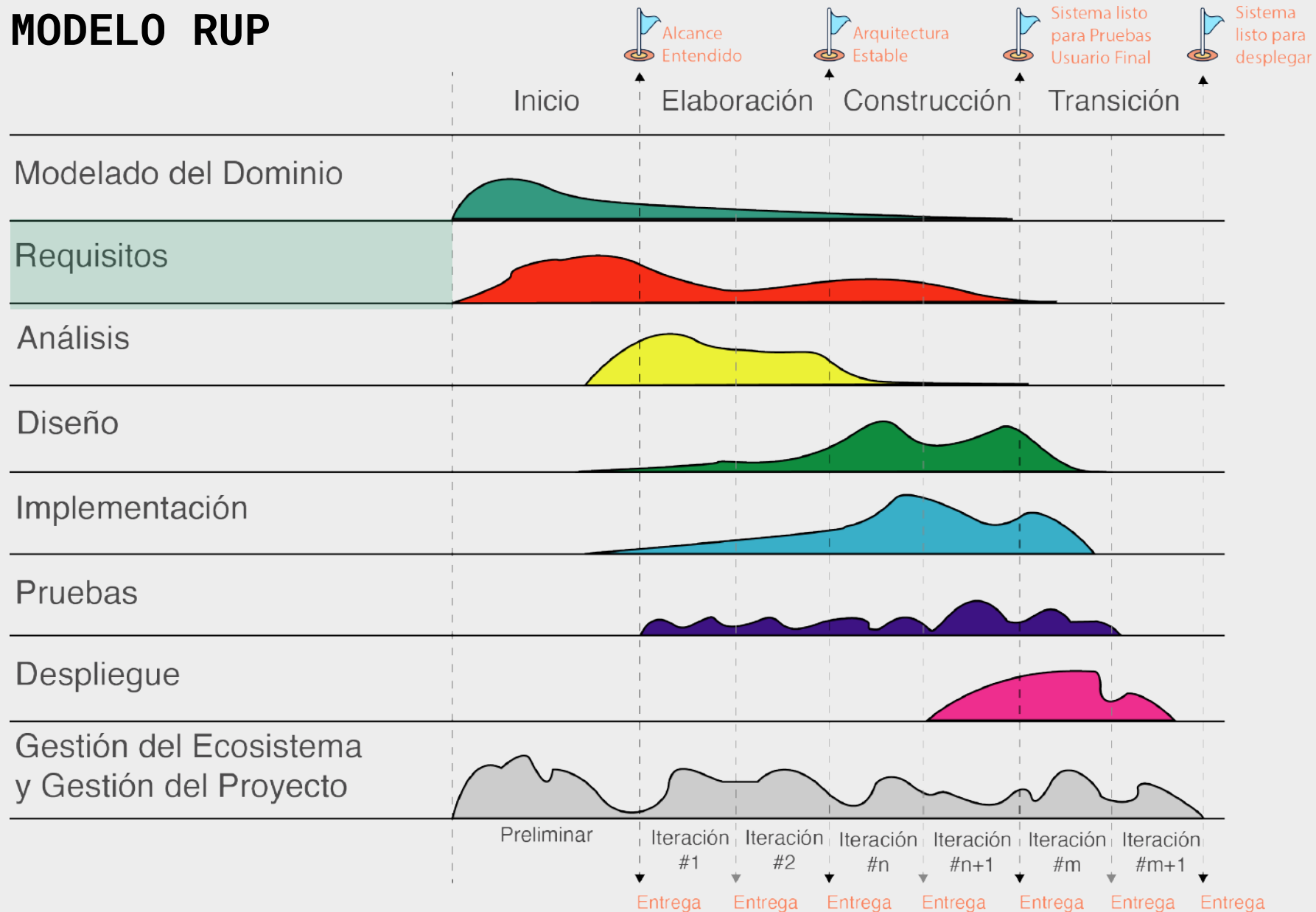
Captura de requisitos

Casos de Uso

Disciplina de Requisitos

Encontrar actores y casos de uso

MODELO RUP

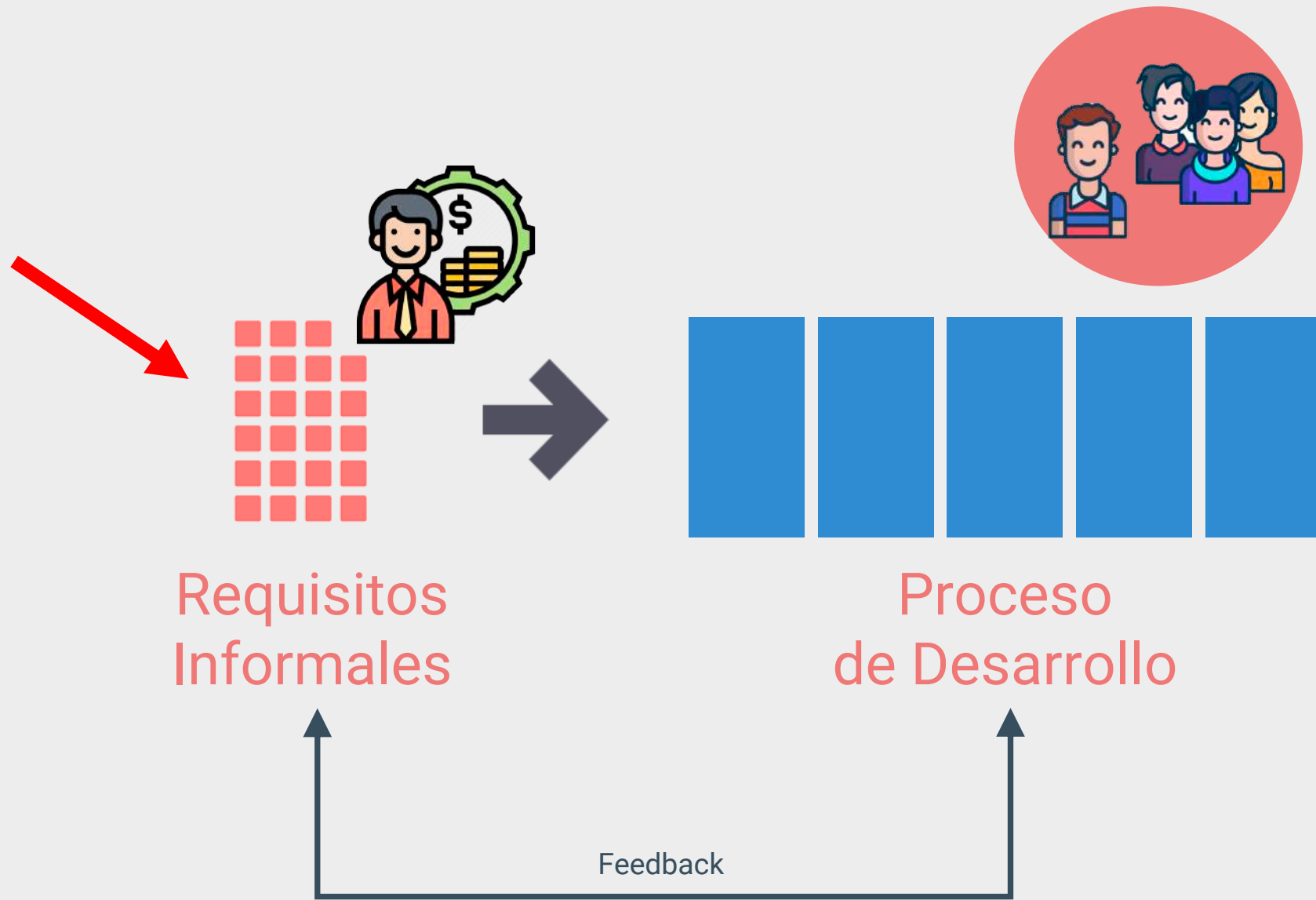


RUP

Nacimiento: 1998

Autor: Booch, Jacobson
y Rumbaugh

- ✓ Flujo iterativo e incremental
- ✓ Pesado
- ✓ Prescriptivo
- ✓ Modelo Iterativo



¿Qué es un requisito?

Un requisito es una **condición o necesidad** de un usuario para resolver un problema o **alcanzar un objetivo**



TIPOS DE REQUISITOS

- Funcionales
- No Funcionales

REQUISITOS FUNCIONALES

Las **condiciones o capacidades** que el sistema debe cumplir

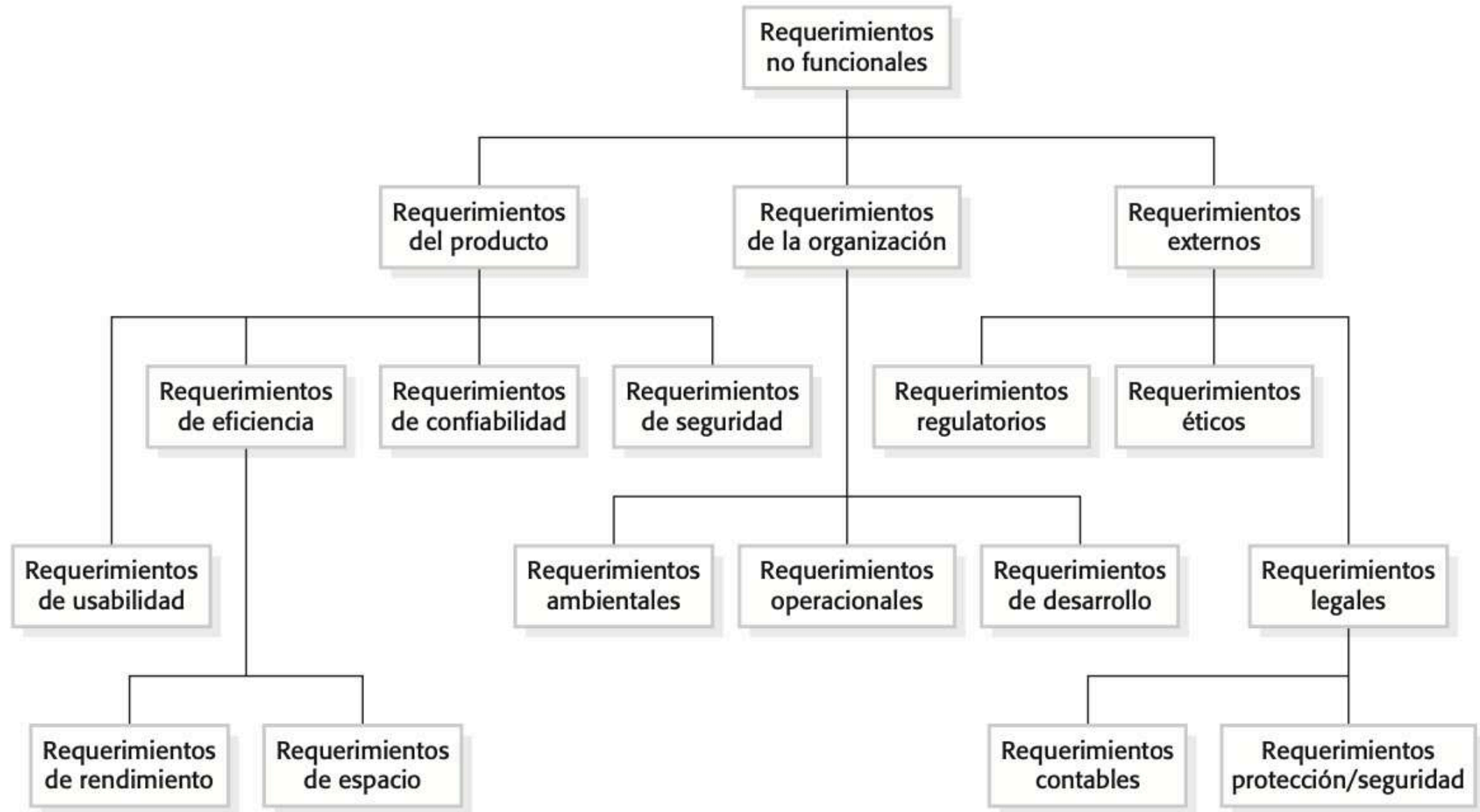
REQUISITOS FUNCIONALES

Definen lo que el sistema debe poder hacer, sin definir cómo se logrará:

- El sistema almacenará la información de contacto
- El sistema aplicará cargos de envío e impuestos al facturar a los clientes

REQUISITOS NO FUNCIONALES

Son **limitaciones** sobre servicios o funciones que ofrece el sistema



RESTRICCIONES

Son limitaciones sobre **cómo se debe implementar una solución** (base de datos, hardware compatible, sistemas operativos, lenguaje de programación)

REQUISITOS NO FUNCIONALES

Afectan más la arquitectura global de un sistema que los componentes individuales

Definen aspectos medibles y característicos de lo que el sistema debe poder hacer

- El sistema devolverá los resultados de la búsqueda en menos de diez segundos
- El sistema deberá procesar al menos 10.000 pedidos por minuto
- Se debe usar la base de datos MySQL

REQUISITOS NO FUNCIONALES

Son de importancia crítica, son distintos en su estructura y contenido y esa distinción debe permanecer, aunque solo sea para ayudar a las personas a concentrarse en redactar buenos requisitos

¿Qué es un
Sistema Software?

¿QUE ES UN SISTEMA DE SOFTWARE?

- ¿Un sistema software es el código máquina, los ejecutables?
- ¿Un sistema software es el código fuente?
- ¿Los subsistemas, clases, diagramas de interacción, diagramas de estado y otros artefactos son el sistema?
- ¿Los requisitos, pruebas, venta, producción, instalación y operaciones son el sistema?

Un sistema de software es **algo** dentro del negocio
que requiere automatizarse

Un sistema es **todos los artefactos** que se necesitan **para representarlo** en una forma comprensible por máquinas u hombres

CAPTURA DE REQUISITOS

La captura de requisitos es un **acto de descubrimiento**,
proceso de averiguar, en circunstancias difíciles,
lo que se debe construir

Las semillas de los desastres enormes del software
por lo general se vislumbran en los tres primeros meses
del inicio del proyecto

— Coper Jones

MOTIVOS DE PROYECTOS FRACASADOS/PROBLEMÁTICOS

Motivo	Incidencia
<ul style="list-style-type: none">• Gestión<ul style="list-style-type: none">• Falta de involucración del usuario• Poco apoyo de las gerencias involucradas• Falta de recursos• Tiempos poco realistas	<ul style="list-style-type: none">• 31.0%<ul style="list-style-type: none">• 12.8%• 7.5%• 6.4%• 4.3%
<ul style="list-style-type: none">• Requisitos<ul style="list-style-type: none">• Requerimientos y especificaciones poco claras• Cambio de requerimientos y especificaciones• Expectativas poco realistas• Objetivos poco claros	<ul style="list-style-type: none">• 35.3%<ul style="list-style-type: none">• 12.3%• 11.8%• 5.9%• 5.3%
<ul style="list-style-type: none">• Tecnologías<ul style="list-style-type: none">• Tecnología deficiente• Nuevas tecnologías	<ul style="list-style-type: none">• 10.7%<ul style="list-style-type: none">• 7.0%• 3.7%
<ul style="list-style-type: none">• Otros	<ul style="list-style-type: none">• 23.0%



— Estadísticas sobre 50.000 proyectos, Chaos Report 1995



Los Requerimientos
del Cliente



Como el Consultor de
Negocios lo describió



Como lo entendió el
Líder del Proyecto



Lo que el Cliente
realmente necesitaba

Defecto de los requisitos	Problemas generados
Implicación insuficiente del cliente	Problemas en la validación del producto obtenido
Requisitos crecientes y cambiantes	Degradación de la estructura y arquitectura del producto
Requisitos ambiguos	Pérdida de tiempo en re-codificación
Requisitos innecesarios	Pérdida de tiempo en trabajo innecesario
Requisitos insuficientes	Problemas en la validación del producto obtenido y/o error en la estimación y planificación
Omisión de las necesidades de grupos de usuarios	Usuarios insatisfechos

Requisito	Especificación
Necesario	Completa
Apropiado	Consistente
No ambiguo	Factible
Completo	Comprensible
Singular	Validable
Verificable	
Correcto	

CARACTERÍSTICA DE UN BUEN REQUISITO

Necesario: define capacidad, característica, restricción y/o factor de calidad esencial

Apropiado: la intención específica y la cantidad de detalle es apropiado para los involucrados

No ambiguo: debe tener una única interpretación

Completo: describe suficientemente la capacidad, características, restricción o factor de calidad sin necesidad de utilizar otra información

Singular: solo involucra una capacidad, característica, restricción o factor de calidad

Verificable: está estructurado y redactado de manera que su realización pueda verificarse, a satisfacción del cliente

Correcto: es una representación precisa de la necesidad de la entidad a partir de la cual se transformó

CARACTERÍSTICA DE UN BUEN CONJUNTO DE REQUISITOS

Completo: describe suficientemente las capacidades, características, restricciones o factores de calidad necesarios para satisfacer las necesidades de la entidad sin necesidad de más información

Consistente: contiene requisitos individuales que son únicos, no entran en conflicto ni se superponen con otros requisitos del conjunto, y las unidades y los sistemas de medición son homogéneos

Factible: se puede realizar dentro de las restricciones de la entidad (por ejemplo, costo, cronograma, técnico) con un riesgo aceptable

Comprensible: está redactado de forma que quede claro lo que espera la entidad y su relación con el sistema del que forma parte

Validable: es factible que la satisfacción del conjunto de requisitos conduzca al logro de las necesidades de la entidad dentro de las limitaciones (por ejemplo, costo, cronograma, cumplimiento técnico, legal y regulatorio)

OBJETIVO DE LA INGENIERÍA DEL SOFTWARE

Los **requisitos** deben ser los **adecuados y necesarios**, y deben mostrar una **visión clara** del **producto final**

FLUJO DE TRABAJO ARQUETÍPICO DE LA CAPTURA DE REQUISITOS

1. Comprender el contexto del sistema (Modelo del dominio)
2. Identificar la visión del producto y los stakeholders que el sistema apoyará
3. Enumerar los requisitos candidatos (lista de características)
4. Capturar requisitos funcionales
5. Capturar requisitos no funcionales

Visión del producto

VISION DEL PRODUCTO

- Nos indica hacia dónde tenemos que ir y qué tenemos que desarrollar
- Transmite un sentido de propósito y de intención
- Nos marca lo que NO debemos desarrollar
- La visión debe enfocarse en el valor para el Cliente
- Indica las características clave del sistema desde la perspectiva de los interesados

NombreClave	Frase representativa Frase corta y precisa que resuma qué es lo que aspira ser el producto		
Grupo de Usuarios	Necesidades	Producto	Valor
<p>Grupo de usuarios que utilizarán el producto. Segmento de mercado al cual nos dirigimos</p>	<p>¿Cuáles son las necesidades del grupo de usuarios?</p>	<p>¿Qué características del producto satisfacen las necesidades de tus usuarios?</p>	<p>¿Qué valor le genera al negocio satisfacer la necesidad del grupo de usuarios a través del producto?</p>

PRODUCT VISION BOARD

GYMSOFT

Sistema para controlar el acceso de las personas a las Instalaciones del Gimnasio.

GRUPO DE USUARIOS

Administrador

Clientes

Empleados

Empleados Especiales

Empresa (Externa)

NECESIDADES

Informe Balance General

Gestionar Sedes

Reserva de clases

Verificación del Plan

Renovar Plan

Validar que un usuario si tenga cuenta Activa al Ingresar al Gym

Crear los horarios de las clases

Vender Planes

Crear Historias Clinicas

Aplicar Descuentos de los convenios

Pagar los planes de los empleados

Registrar Beneficiarios

FUNCIONALIDADES

Consultar Usuarios Activos.

Crear Sedes.

Seleccionar Clase

Consultar Plan Vigente

Pago en Linea

Consultar Plan Vigente de Usuario

Crear Clase

Generar Facturas

Registrar Datos del Paciente

Consultar Datos del Paciente

Crear Beneficiarios

Tipificar Empleados

BENEFICIOS

Información en tiempo Real

Información Segmentada y Organizada

Acceder a la información desde cualquier lugar

Ahorro de tiempo

Evitar el fraude con los ingresos

Control de la salud del paciente

Convenios con empresas externas

Es un proyecto de las personas del gimnasio que se llama GYM y aplicación frente a los otros mercados.

El...

BD web

Empleado

Mapa

Vision

aplicación para compartir fotografías creativas y únicas

Target Group

Mujer entre
18-29 años,
urbana, con
estudios
universitarios

Hombre
Hispano,
suburbano,
con algunos
estudios

Gente Joven
(la generación
millennial y la
que le sigue)

Gente con
ganas de
viajar, de
probar nuevos
productos,

Needs

Compartir
fotos y
videos con
los amigos

Publicitar un
producto
entre los
usuario

Seguir las
publicaciones
de amigos

Conocer
Gente

Aplicar filtros
especiales, a
las imagenes
para destacar

explotar los
elementos
principales de la
fotografía para
que los usuarios
se comuniquen

Product

Crear
usuarios y
mantener
perfiles

Subir videos
y fotos y
publicarlos

Etiquetar
personas o
colocar
Hashtags

Dar opinion
sobre las
publicaciones
(me gusta)

Aplicar filtros
en vivo

Etiquetar
Productos y
redirigir a
sitio de
compra

Business Goals

Obtener más
usuarios para
poder
mostrarles
publicidad

Atraer
empresas
para que
publiciten sus
productos

Más
usuarios,
más
ingresos

Más
usuarios,
más
publicidad

Stakeholders

STAKEHOLDERS

- Todos los participantes (interesados o afectados) por un proyecto del lado del cliente
- Cualquier persona que se beneficie en forma directa o indirecta del sistema en desarrollo



Business Owner



End Users



Domain Expert

STAKEHOLDERS

El CEO de la empresa

Gerentes de operaciones del negocio

Gerentes de producto

Personal de mercadotecnia

Clientes internos y externos

Consultores

Ingenieros de producto

Ingenieros de software

Personal de apoyo y mantenimiento

Expertos en el dominio

Empleados

Diseñadores gráficos

Usuarios finales

PUNTOS DE VISTA



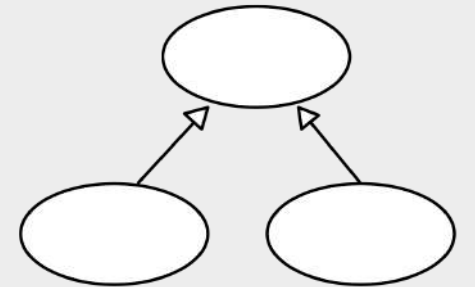
Técnicas

TÉCNICAS LEVANTAMIENTOS DE REQUISITOS

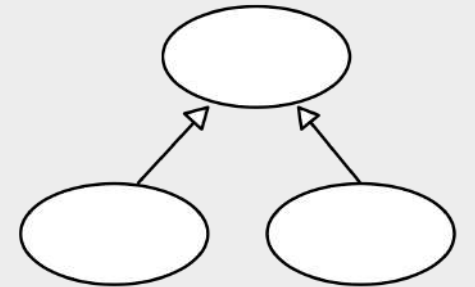
- Casos de Uso
- Historias de Usuarios
- Prototipos (Prototipo de Interfaz)

CASOS DE USO

Los casos de uso son **fragmentos de funcionalidad** que el sistema ofrece para aportar un resultado de valor para sus actores



Es una especificación de **secuencias de acciones**,
incluyendo variaciones, que el sistema puede realizar
y que dan un **resultado observable de interés**
a un actor particular



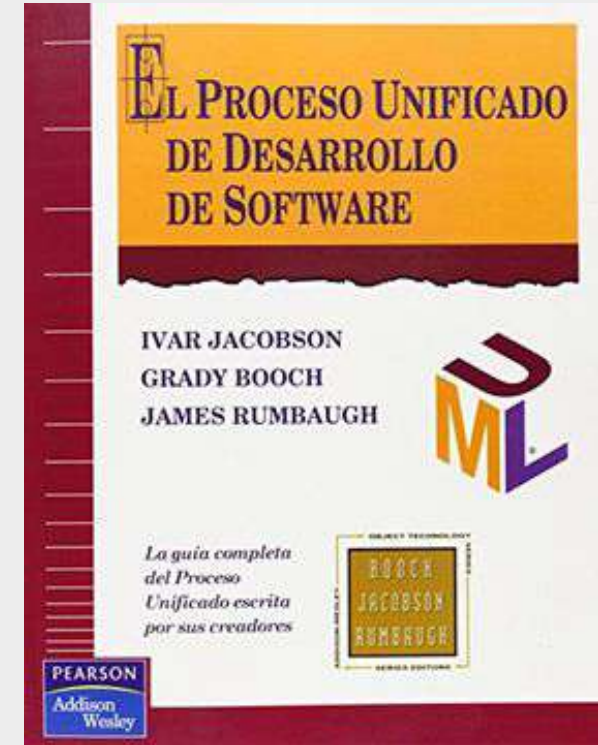
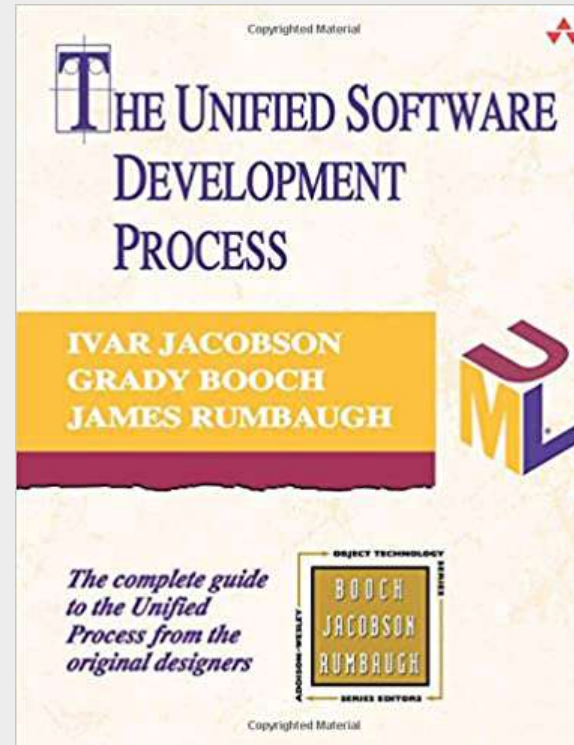
CASOS DE USO



Ivar Jacobson

Ericsson

Método de Desarrollo:
Objectory

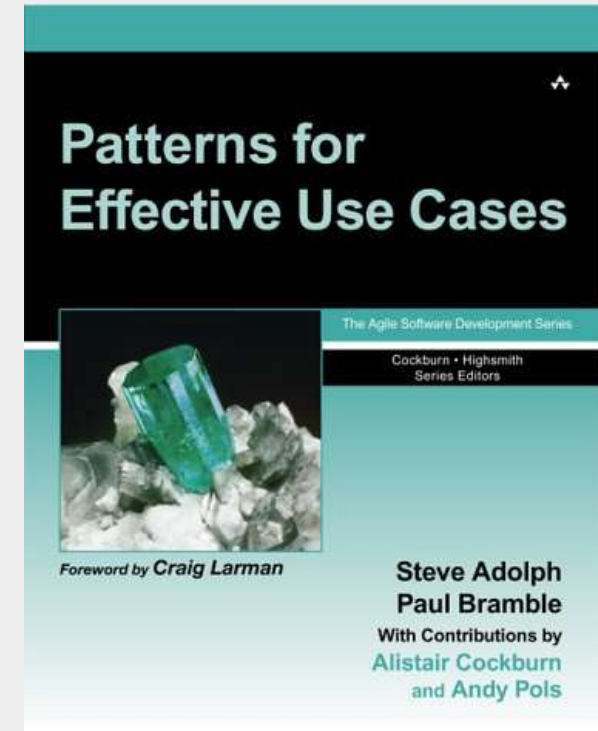
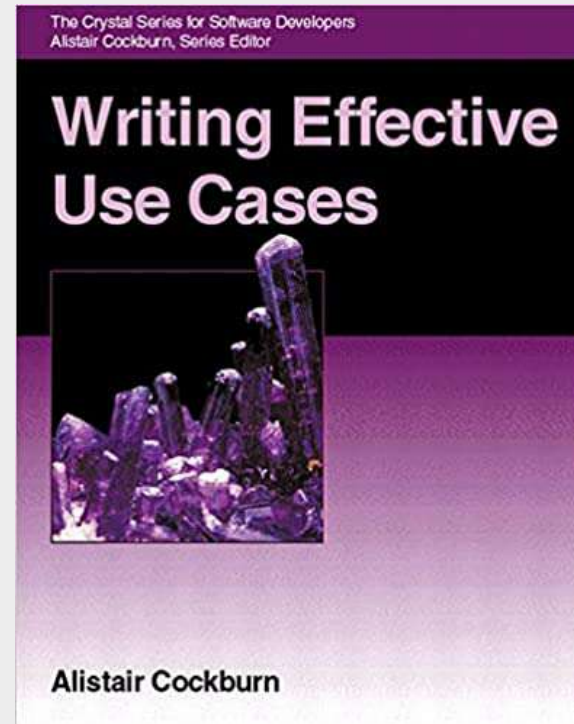


CASOS DE USO



Alistair Cockburn

Crystal family: Crystal Clear



Heart of Agile

Collaborate • Deliver • Reflect • Improve

CASOS DE USO

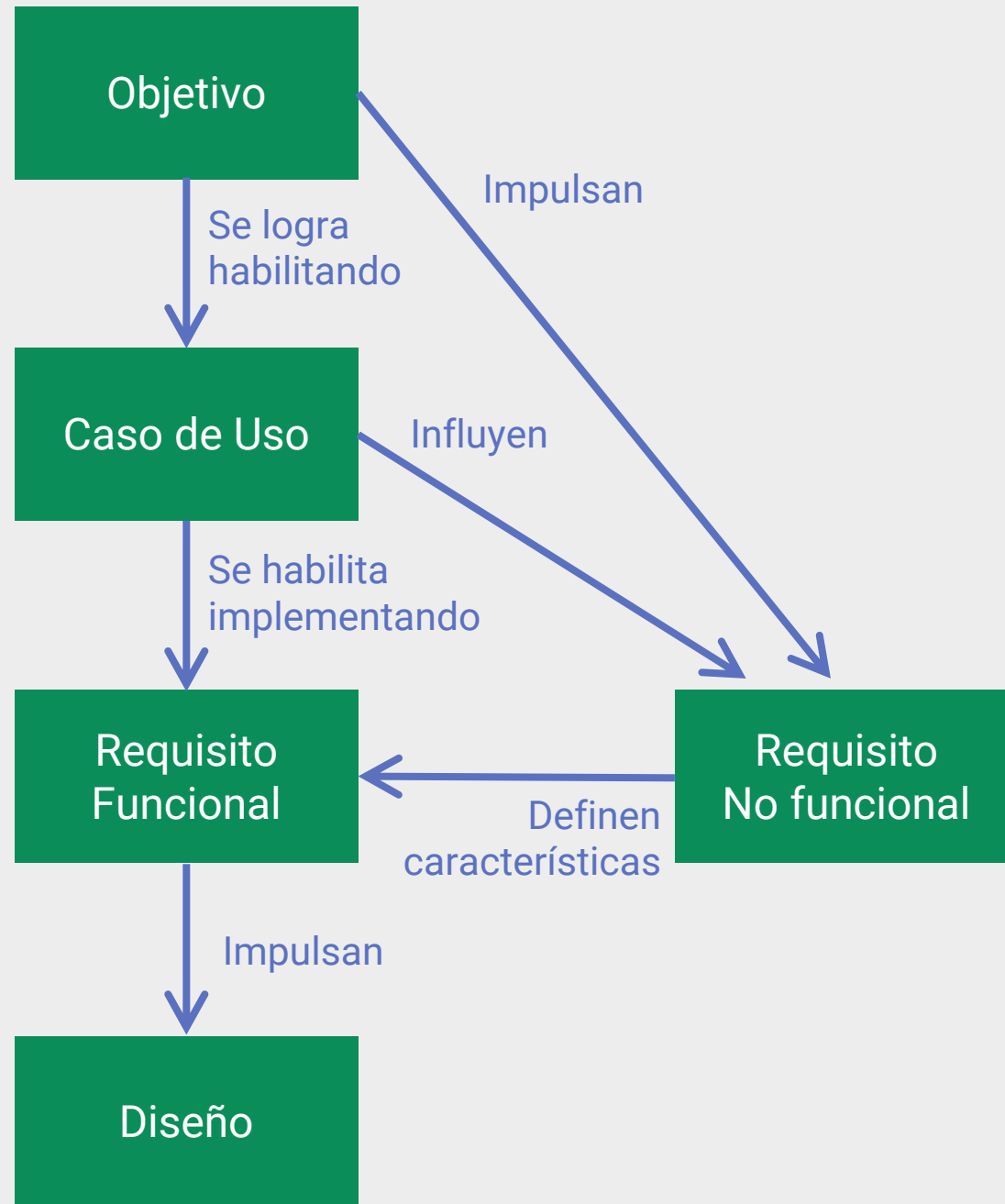
- Son una forma adecuada de crear un modelo de sistema
- Proporcionan un medio intuitivo y sistemático para capturar los requisitos
- Cuentan historias coherentes sobre cómo se comportará el sistema en uso
- Obligan a pensar en términos de quiénes son los usuarios y qué necesidades u objetivos de la empresa pueden cumplir

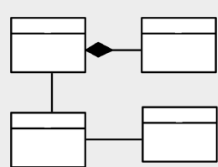
CASOS DE USO

- Los requisitos funcionales se estructuran de forma natural con ellos
- La mayoría de los requisitos no funcionales son específicos de un sólo caso de uso y pueden tratarse en el contexto de ese caso de uso

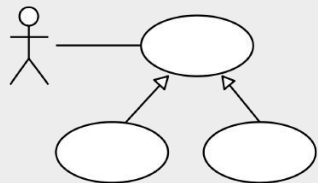
CASOS DE USO COMO REQUISITOS

- **Son requisitos:** detallan con precisión lo que debe hacer el sistema
- **Pero no todos los requisitos:** no detallan interfaces externas, formatos de datos, reglas comerciales y fórmulas complejas

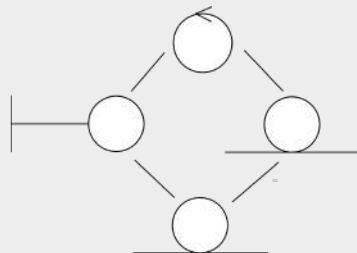




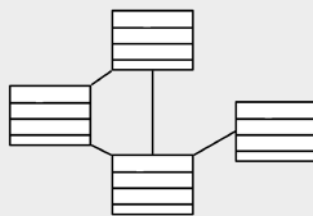
Modelo de Dominio



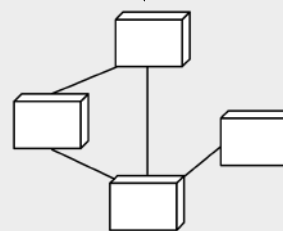
Modelo de Casos de Uso



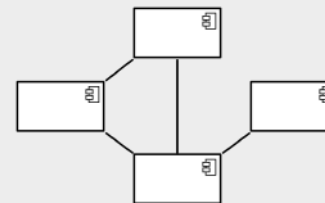
Modelo de Análisis



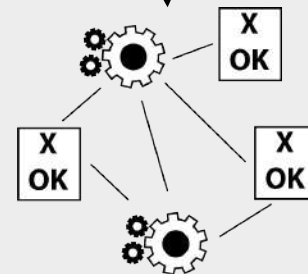
Modelo de Diseño



Modelo de Despliegue



Modelo de Implementación



Modelo de Prueba



Especificado por

Realizado por

Distribuido por

Implementado por

Verificado por

DISCIPLINA DE REQUISITOS

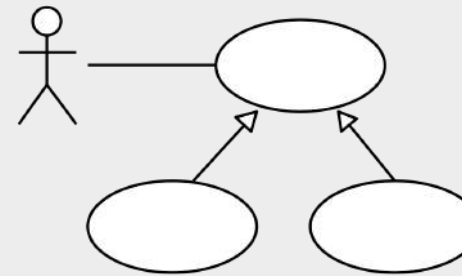
La **disciplina de requisitos** busca describir los requisitos del **sistema** para poder alcanzar un acuerdo entre los clientes, usuarios y desarrolladores sobre lo que el sistema debería hacer

Disciplinas Técnicas	Disciplinas de Soporte
Establecer y mantener el acuerdo con los clientes y otros implicados sobre lo que el sistema debería hacer	Proveer las bases para planificar los contenidos técnicos de cada iteración
Proveer a los desarrolladores del sistema con una comprensión de los requisitos del sistema	Proveer las bases para la estimación de costes y tiempos para desarrollar el sistema
Definir los límites del sistema	

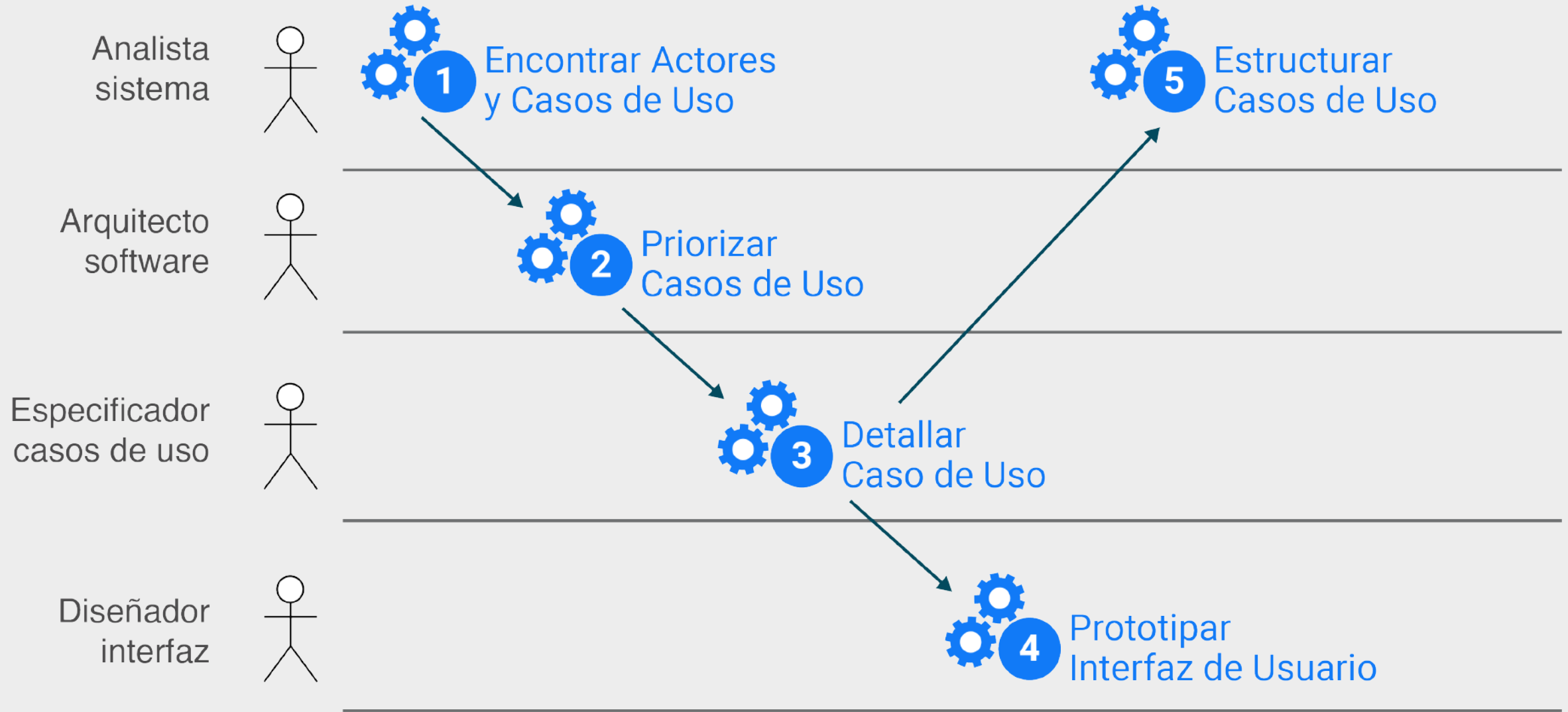
Disciplina de Requisitos:

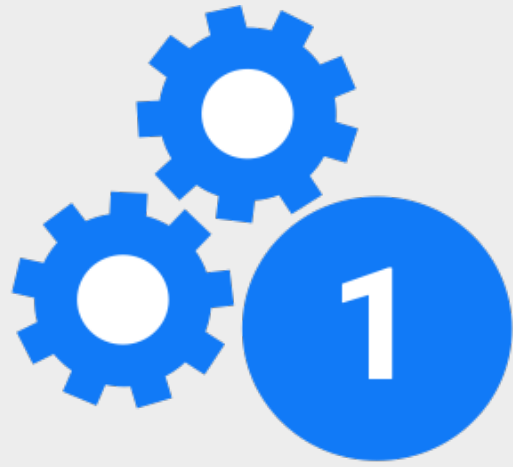
1. Encontrar Actores y Casos de Uso
2. Priorizar Casos de Uso
3. Detallar Caso de Uso
4. Prototipar Interfaz de Usuario
5. Estructurar Casos de Uso

→ **Modelo de Casos de Uso**

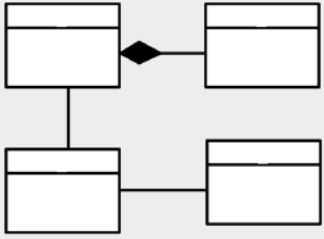


- ✓ Diagrama de Casos de Uso
- ✓ Especificaciones de Casos de Uso
- ✓ Diagrama de contexto
- ✓ Prototipo de Interfaz





Encontrar Actores y Casos de Uso



Modelo de Dominio



Lista de
Características

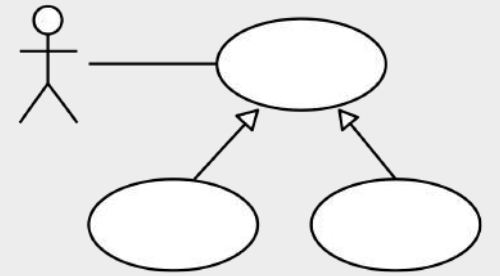


Requisitos
no funcionales



1

Encontrar Actores
y Casos de Uso



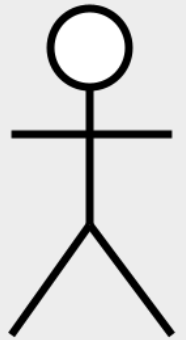
Modelo de Casos
de Uso (esbozo)

PASOS

- Encontrar actores
- Encontrar los casos de uso
- Describir brevemente cada caso de uso
- Describir el modelo de casos de uso completo

Encontrar actores

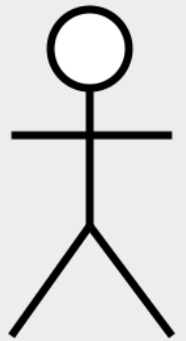
Los actores representan **terceros fuera del sistema**
que **colaboran con el sistema**



ENCONTRAR ACTORES DEL SISTEMA

- ¿Quién o qué usa el sistema?
- ¿Quién obtiene y provee información al sistema?
- ¿Qué roles juegan en la interacción?

Un actor puede ser una persona, una empresa u organización, un programa informático o un sistema informático: hardware, software o ambos

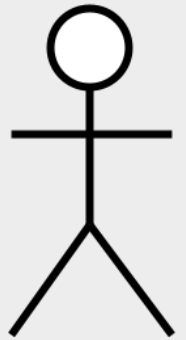


TIPO DE ACTORES DEL SISTEMA

- Un **actor principal** es el aquel que inicia un Caso de Uso y obtiene un beneficio cuando se logra el propósito del mismo
- Un **actor secundario** es aquel que participa en el flujo de eventos de un Caso de Uso, pero nunca lo comienza

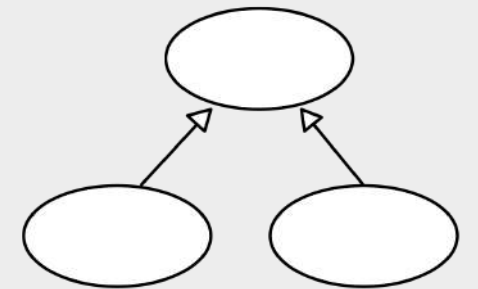
Actor Principal	Actor Secundario
Utiliza las funcionalidades del sistema	Depende de las acciones que realiza el actor primario
Accede directamente al sistema	Solo participa de la interacción para obtener el propósito
Interactúan directamente con el sistema	

Actor tiempo se usa para modelar cosas que suceden en un punto específico del tiempo, pero no parecen ser lanzados por ningún actor

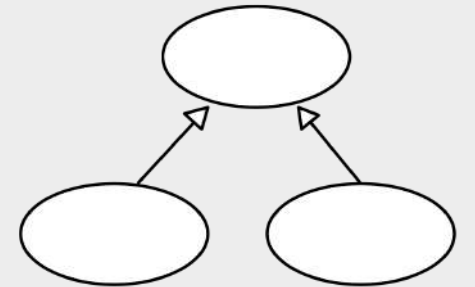


**Encontrar los
casos de usos**

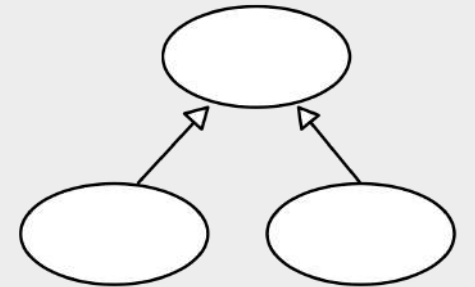
El actor típicamente necesitará **casos de uso**
para **soportar su trabajo**



El actor necesitará crear, modificar, borrar
o leer **objetos de negocio**



Un caso de uso entrega un **resultado que se puede observar** y que **añade valor** a un actor en concreto



ENCONTRAR CASOS DE USO DEL SISTEMA

- ¿Cuáles son los objetivos de los actores?
- ¿Qué tareas o funciones principales son realizadas por el actor?
- ¿Qué información del negocio adquiere, produce o cambia el actor?
- ¿Qué información desea obtener el actor?

ENCONTRAR CASOS DE USO DEL SISTEMA

- Los nombres de los casos de uso deben comenzar con un **verbo** y **reflejar el objetivo** de la interacción entre el actor y el sistema
- Los casos de uso deben ser **atómicos**: su comportamiento es independiente de otro caso de uso y debe ejecutarse por completo o no ejecutarse como una **unidad**

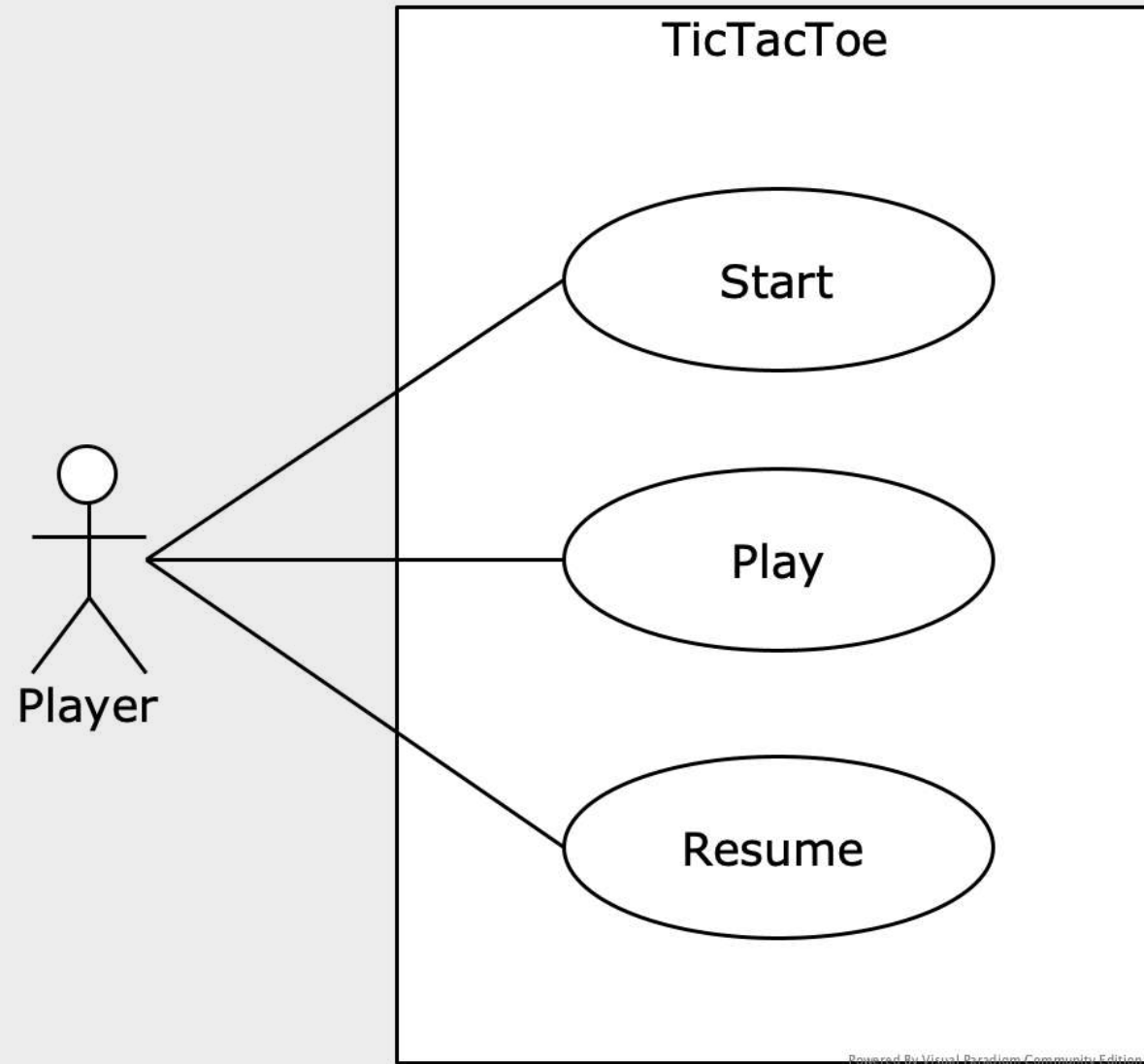
Describir

DESCRIBIR BREVEMENTE CADA CASO DE USO

- Explicar en pocas palabras que hace cada caso de uso si es necesario
- Usar algunas frases que resumen las acciones

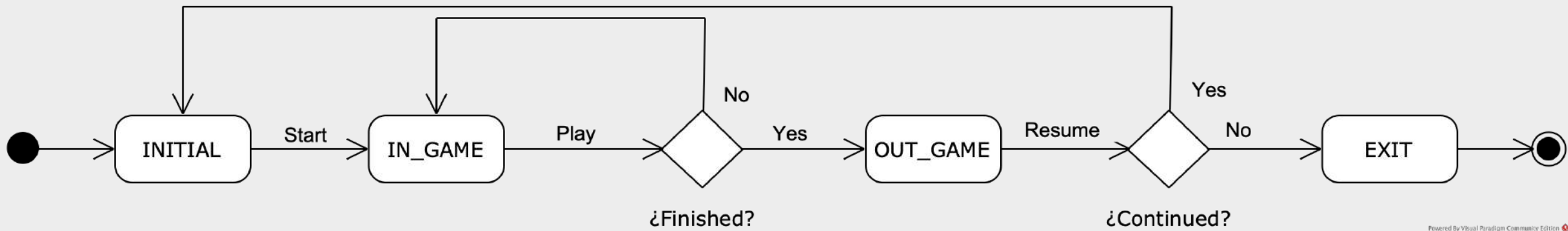
DESCRIBIR EL MODELO DE CASOS DE USO COMPLETO

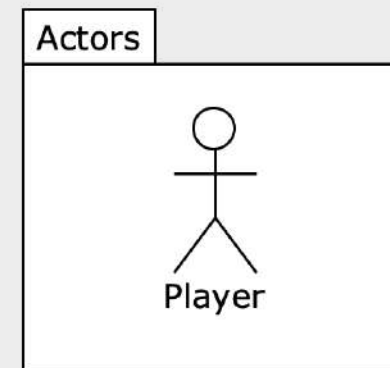
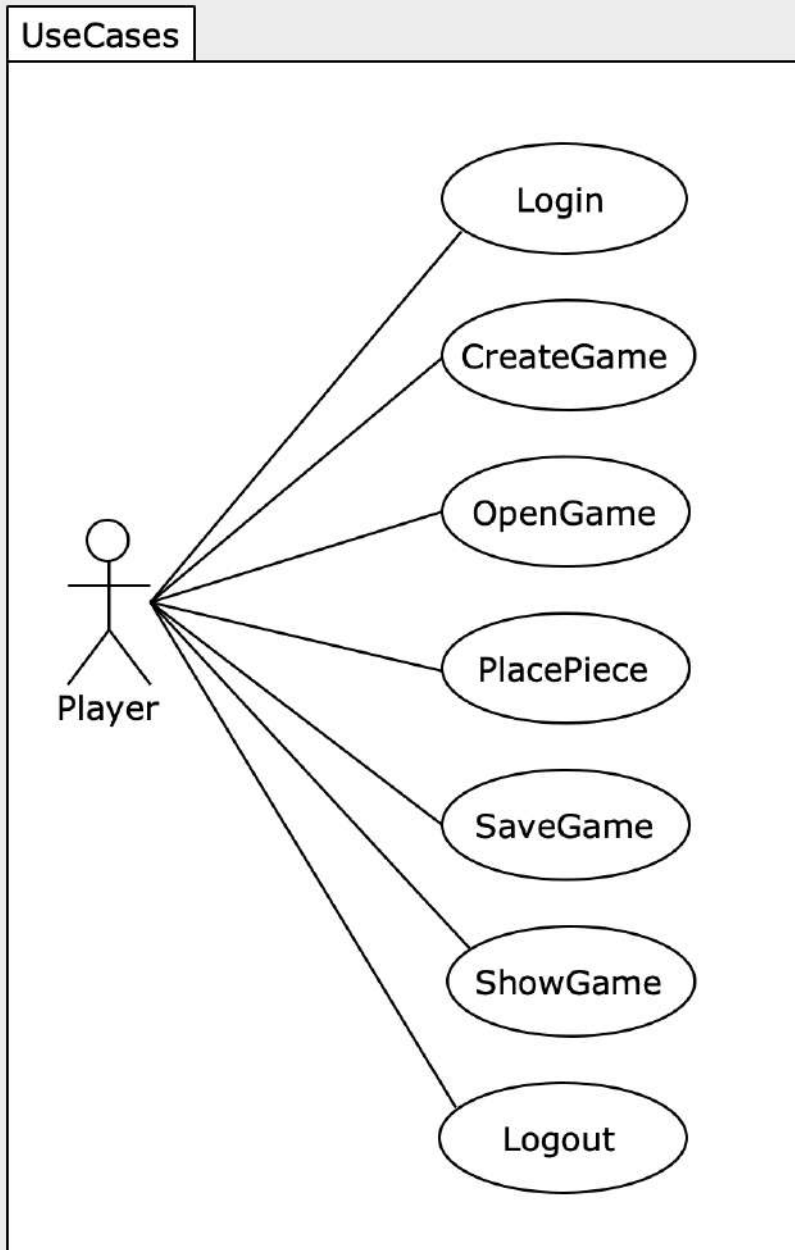
- Explicar como los actores se relacionan con los casos de uso usando un **diagrama de casos de uso**
- Explicar como los casos de uso se relacionan entre sí usando **diagramas de estado**

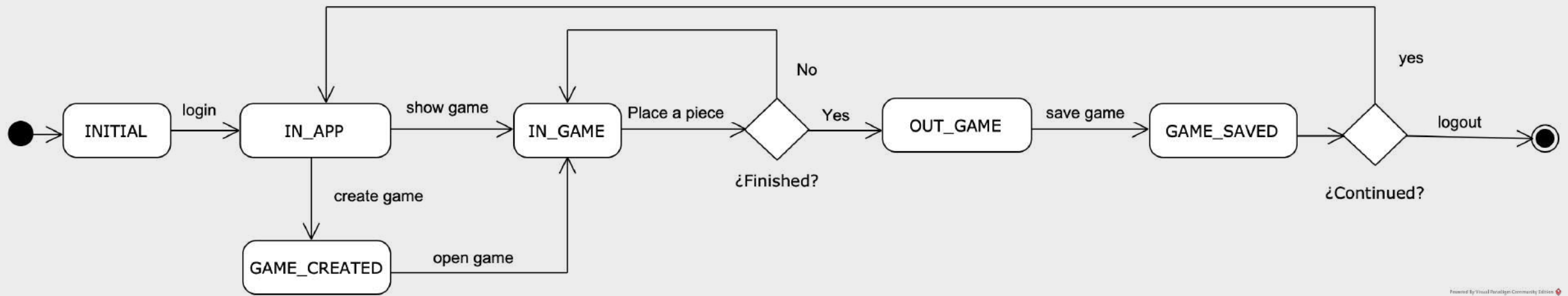


DESCRIBIR EL MODELO DE CASOS DE USO COMPLETO

- **Start:** el jugador solicita comenzar un nuevo juego y el sistema le muestra el título, el tablero vacío
- **Play:** el jugador solita poner o mover fichas, el sistema le permite hacerlo y le va mostrando el resultado
- **Resume:** el jugador solicita reanudar el juego, el sistema le pregunta si quiere o no seguir jugando







Ejemplo

ESCENARIO

- El Consorcio Interbank decide diseñar el Sistema de Pagos y Facturación en colaboración con algunos de sus principales clientes
- La motivación es atraer nuevos clientes ofreciendo comisiones bajas por el proceso de los pagos y reducir sus costes laborales procesando las solicitudes de cambio automáticamente por Internet, en lugar de manualmente mediante cajeros
- Las motivaciones para compradores y vendedores son reducir los costes, el papeleo y el tiempo de proceso

ESCENARIO

- El sistema utilizará Internet para el envío de pedidos, facturas, y pagos entre compradores y vendedores
- Los compradores y vendedores no tendrán que enviar pedidos o facturas por correo ordinario
- El pago de las facturas se llevará a cabo entre los computadores del comprador y del vendedor
- Ambos tendrán también una visión general del estado de sus facturas y pagos

ESCENARIO

- El comprador podrá buscar los productos y precios, realizar un pedido
- Los bienes o servicios le llegarán al comprador junto con una factura
- El comprador podrá dar el visto bueno a la factura recibida y planificar el pago requerido, llegada la fecha planificada se deberá transferir automáticamente el dinero desde la cuenta del comprador a la cuenta del vendedor

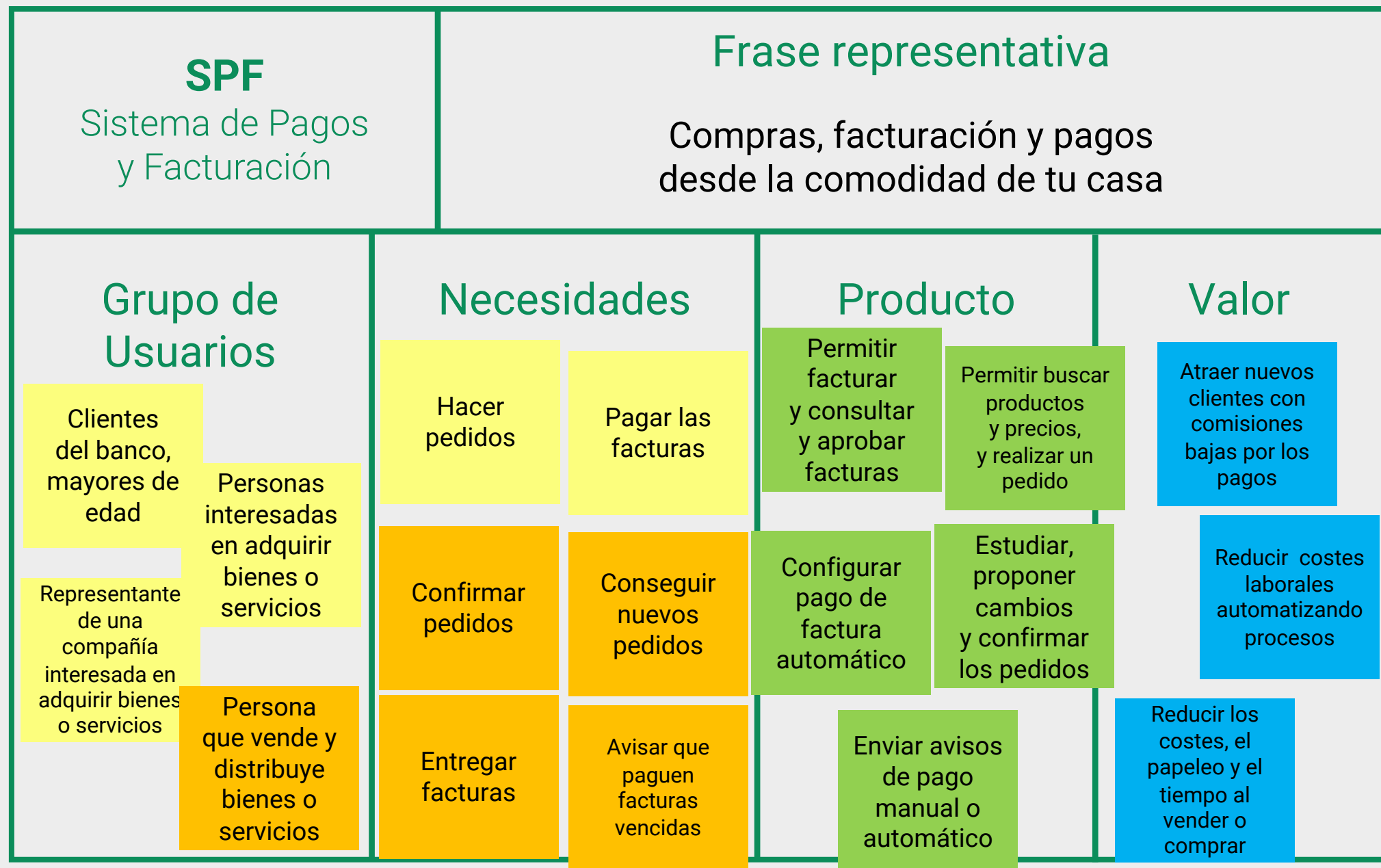
ESCENARIO

- El vendedor puede estudiar, proponer cambios y confirmar los pedidos recibidos
- Cuando confirma un pedido el vendedor entregará los bienes o servicios (esto se hará fuera del Sistema de Facturación y Pagos)
- Cuando hayan sido entregados los bienes o servicios, el vendedor facturará al comprador
- Al llevar a cabo la facturación, el vendedor puede que tenga que aplicar un descuento y puede que también elija combinar varias facturas en una
- Si el comprador no ha pagado en la fecha de vencimiento, se informará al vendedor y éste podrá un aviso al vendedor recordándole el pago
- El sistema podría enviar avisos automáticamente, pero hay que elegir una solución en la que el vendedor tenga la oportunidad de revisar los avisos antes de que sean enviados para evitar violentar al cliente

ESCENARIO

- Cuando un comprador envía una factura para su pago, el sistema debería responder con una verificación de la solicitud en menos de 1.0 segundos
- La duración de esta verificación nunca deberá exceder los 10.0 segundos a menos que la conexión de red no funcione (en cuyo caso se debe informar al usuario)

NombreClave	Frase representativa Frase corta y precisa que resuma qué es lo que aspira ser el producto		
Grupo de Usuarios	Necesidades	Producto	Valor
Grupo de usuarios que utilizarán el producto. Segmento de mercado al cual nos dirigimos	¿Cuáles son las necesidades del grupo de usuarios?	¿Qué características del producto satisfacen las necesidades de tus usuarios?	¿Qué valor le genera al negocio satisfacer la necesidad del grupo de usuarios a través del producto?

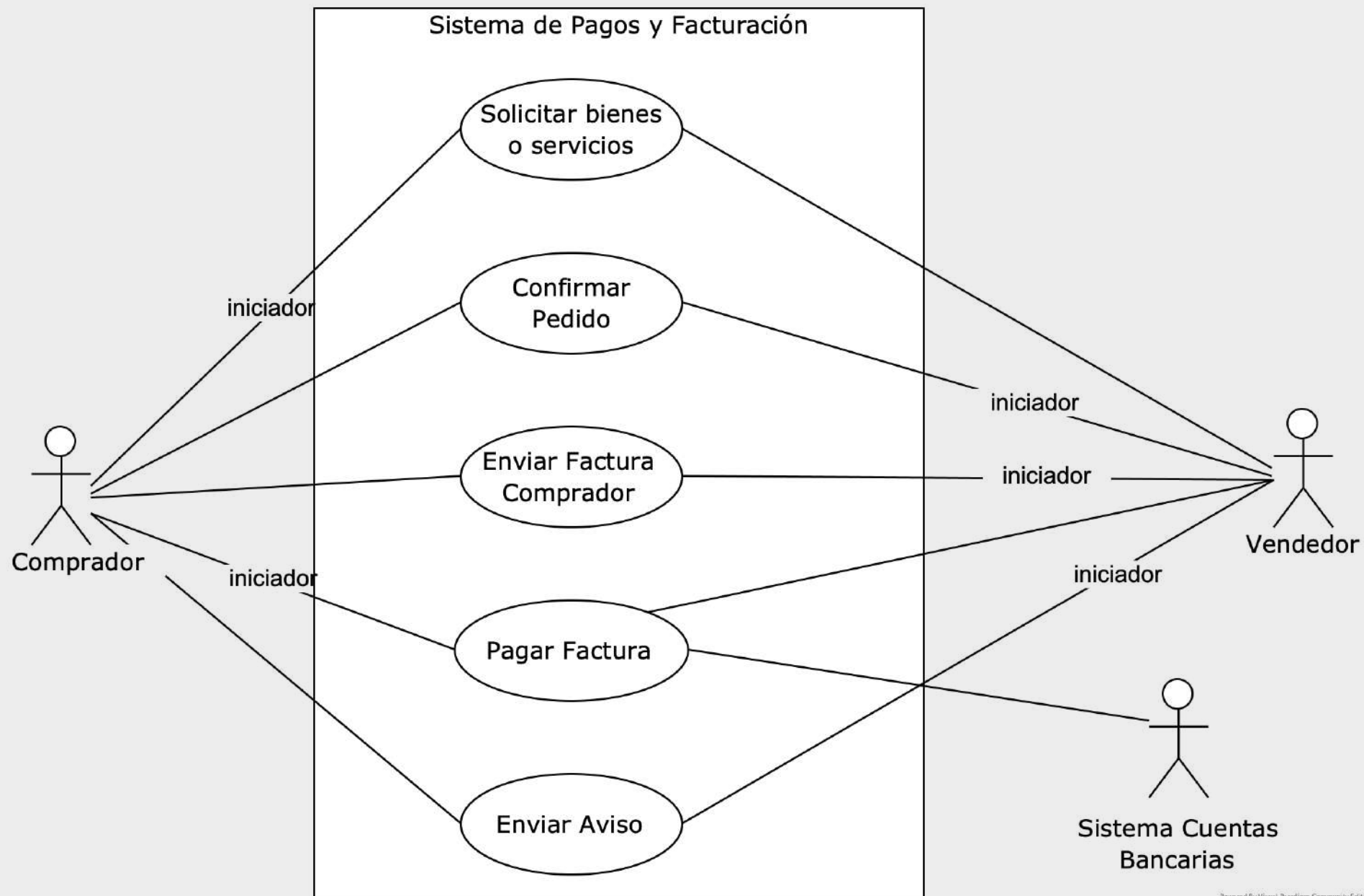


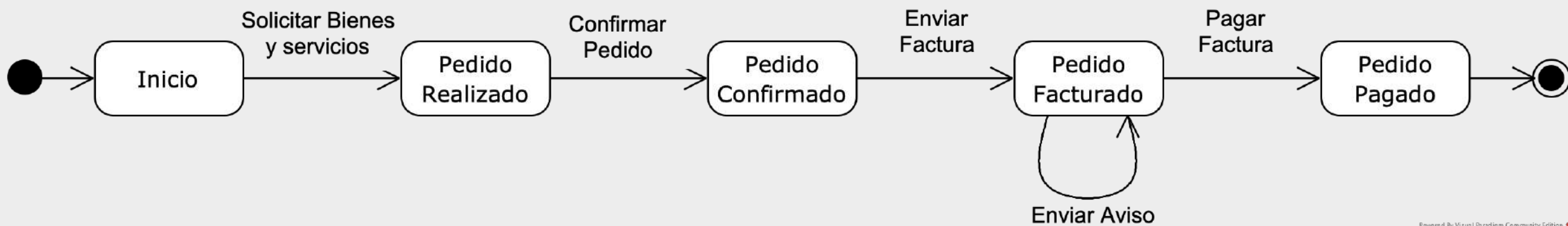
ACTORES

- **Comprador:** representa a una persona que es responsable de adquirir bienes o servicios. Esta persona puede ser un individuo (es decir, no asociado a una compañía), o alguien dentro de una empresa. El Comprador de bienes y servicios necesita el Sistema de Facturación y Pagos para enviar pedidos y pagar las facturas
- **Vendedor:** representa a una persona que vende y distribuye bienes o servicios. El Vendedor utiliza el sistema para conseguir nuevos pedidos y entregar las confirmaciones de pedido, facturas y avisos de pago
- **Sistema de Cuentas Bancarias:** el Sistema de Facturación y Pagos envía verificaciones de transacciones al Sistema de Cuentas Bancarias

CASOS DE USOS

- **Solicitar bienes o servicios:** lo utiliza un comprador para buscar los productos y precios, para realizar un pedido y después enviarlo
- **Confirmar pedido:** lo utiliza el vendedor para estudiar, proponer cambios y confirmar los pedidos recibidos. Un pedido confirmado estará seguido de la entrega de los bienes o servicios
- **Enviar factura comprador:** lo utiliza el vendedor para facturar al comprador. El vendedor puede que tenga que aplicar un descuento y puede que también elija combinar varias facturas en una
- **Pagar factura:** lo utiliza un comprador para planificar los pagos de las facturas por los bienes que él o ella ha solicitado y recibido. Efectúa el pago el día de vencimiento de la factura
- **Enviar aviso:** si el comprador no paga en la fecha de vencimiento, se informará al vendedor y éste podrá usar este caso de uso para enviar Aviso





REFERENCIA

“El Proceso Unificado de Desarrollo de Software”.

Capítulo 6 y 7. Booch, Rumbaugh y Jacobson.
2000. Addison-Wesley

“UML y Patrones”. Capítulo 6, 9 y 13.

Craig Larman, 2da Edición, Prentice-Hall

¡Hasta la próxima clase!