



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

METODOLOGÍA ÁGIL XP

Prácticas ágiles en la metodología XP, criterios de aceptación , estadísticas de uso las metodologías ágiles y BPM

03

PROYECTO

Idea del proyecto del curso

Conceptos

Resumen





Desarrollo de software ágil

Resumen

Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo, en lugar de intentar entregarlo todo de una vez al final del proyecto.





Desarrollo de software ágil

Resumen

Ágil es una forma de pensar y de hacer las cosas y un comportamiento durante un proceso de desarrollo

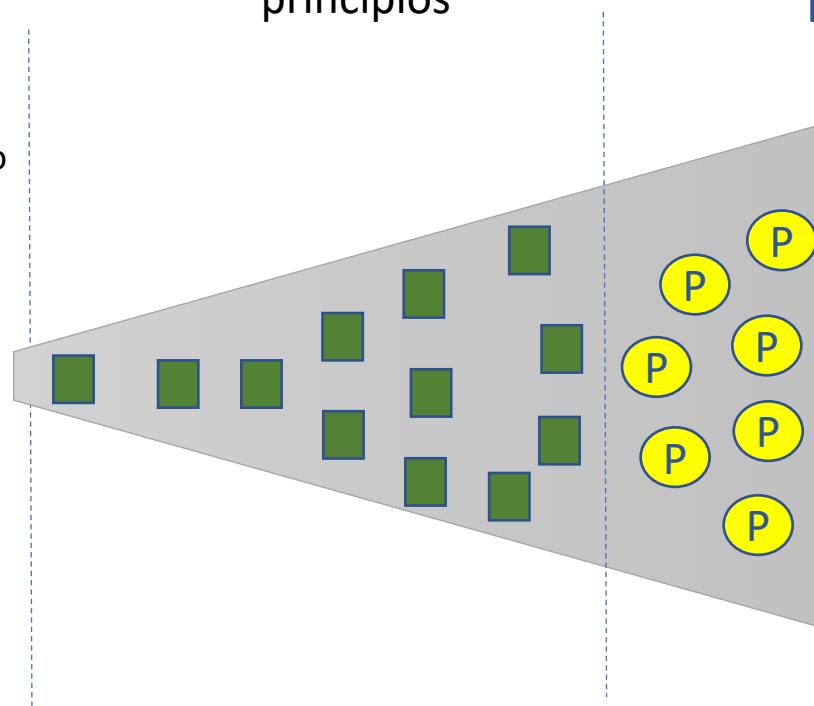
Ágil es una mentalidad



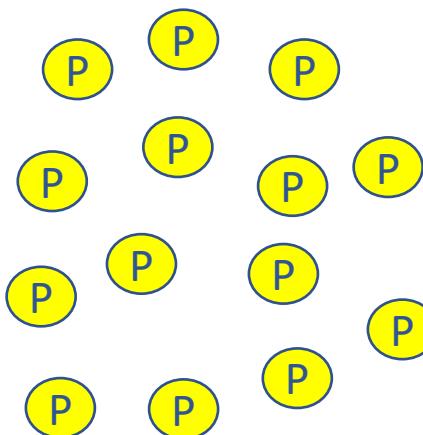
Descripción por 4 valores



Definido por 12 principios



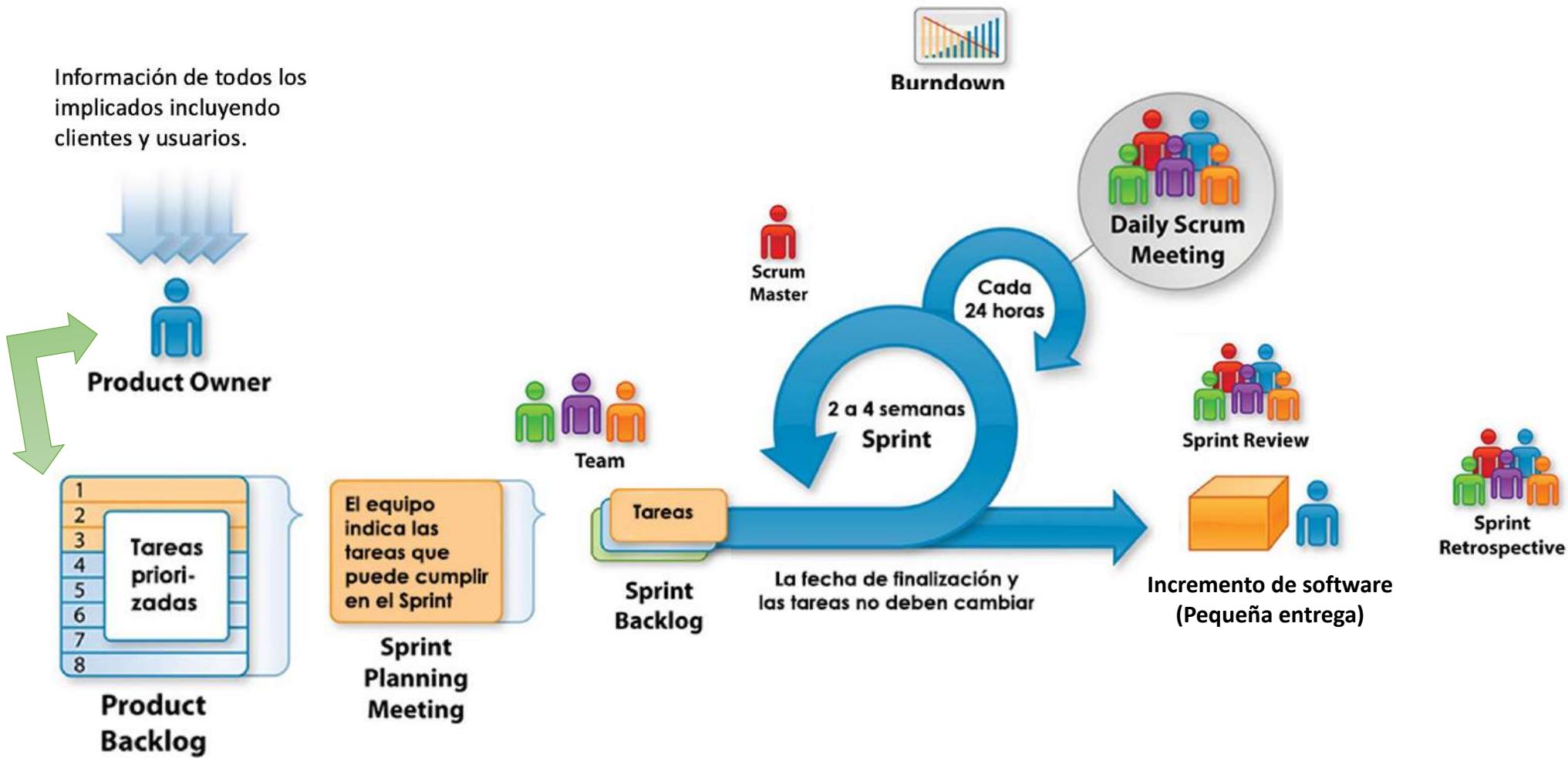
Manifestada a través de un Número ilimitado de prácticas



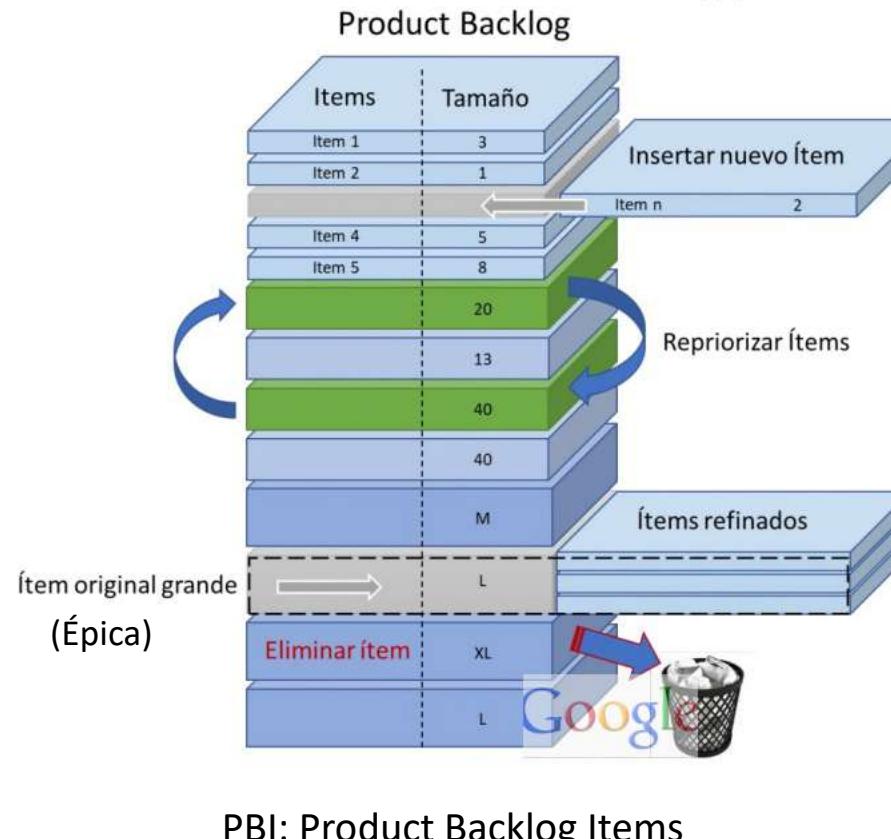
P= práctica ágil

Conceptos

Metodología ágil Scrum

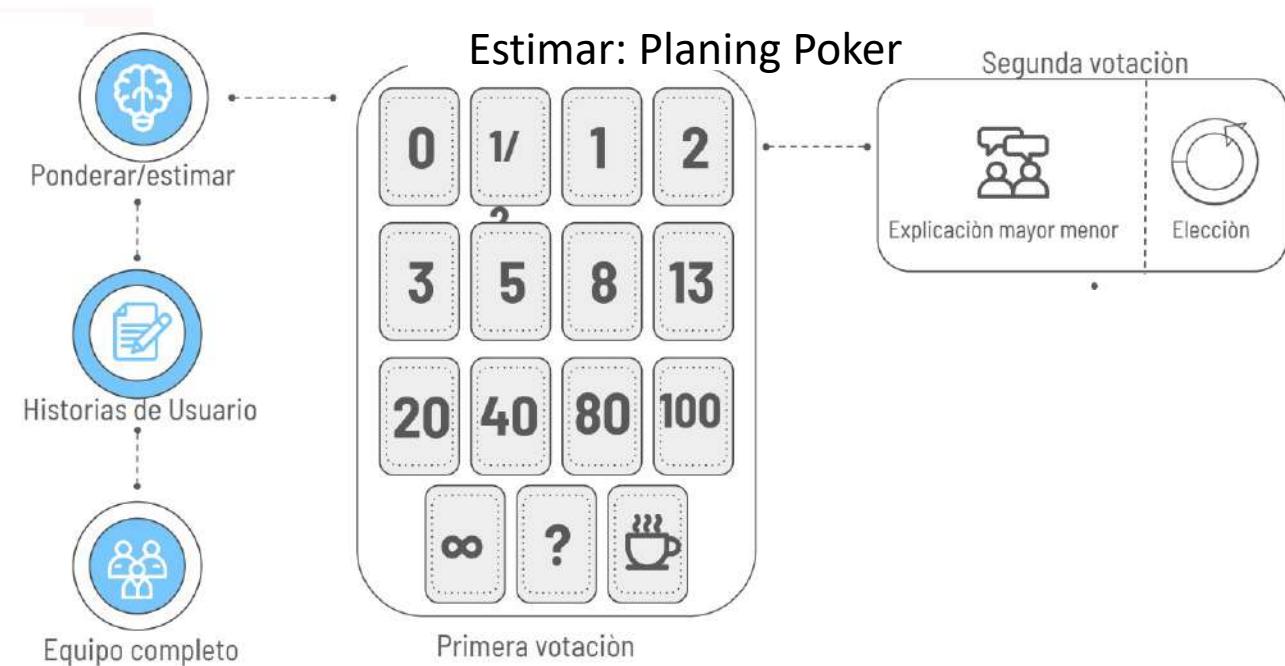


Product Backlog



Características

- El Backlog es un documento “dinámico”.
- Los ítems del Backlog deben agregar siempre valor.
- Los ítems se deben priorizar (Alta, Baja, Media)
- Todos los ítems deben estimarse.



REUNIONES EN SCRUM

Reuniones o ceremonias que se realizan en un Sprint

5 ceremonias Scrum:



Sprint Planning



Daily Scrum



Sprint Review



Sprint Retrospective



Sprint Grooming o Refinement

PRÁCTICAS ÁGILES EN SCRUM

Release Plan

Sprint	HU	Puntos de HU	Desarrollador	Road Map
1	HU1	3	Pedro Juan	
	HU2	4		
2	HU3	2	Maria Pedro/Ricardo Juan	
	HU4	4		
	HU6	2		
3	HU5	4	Maria Juan	
	HU7	3		
4	HU9	3	Pedro Juan	
	HU8	3		
5	HU10	4	Maria Pedro	
	HU11	3		

Velocidad

7

8

7

6

7

Release 1

Release 2

Release 3

Velocidad: Cantidad de puntos que un equipo es capaz de hacer en un sprint



Preguntas ?





Metodología Ágil XP



**Actividad de evaluación
Se aplaza para la próxima
clase**

Metodología Ágil XP

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» **Kent Beck**.

¿Qué es XP?

- La Programación Extrema es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

Metodología Ágil XP

- XP surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos
- XP es orientada principalmente a la codificación
- Programación basada en los “deseos” del cliente.
- Los participantes en un proyecto lo conforman los jefes de proyecto, desarrolladores, el cliente y otros (stakeholders)
- Se rige por valores y principios definidos en el manifiesto ágil



Metodología Ágil XP

Actividades

Codificación: La parte mas importante de XP.

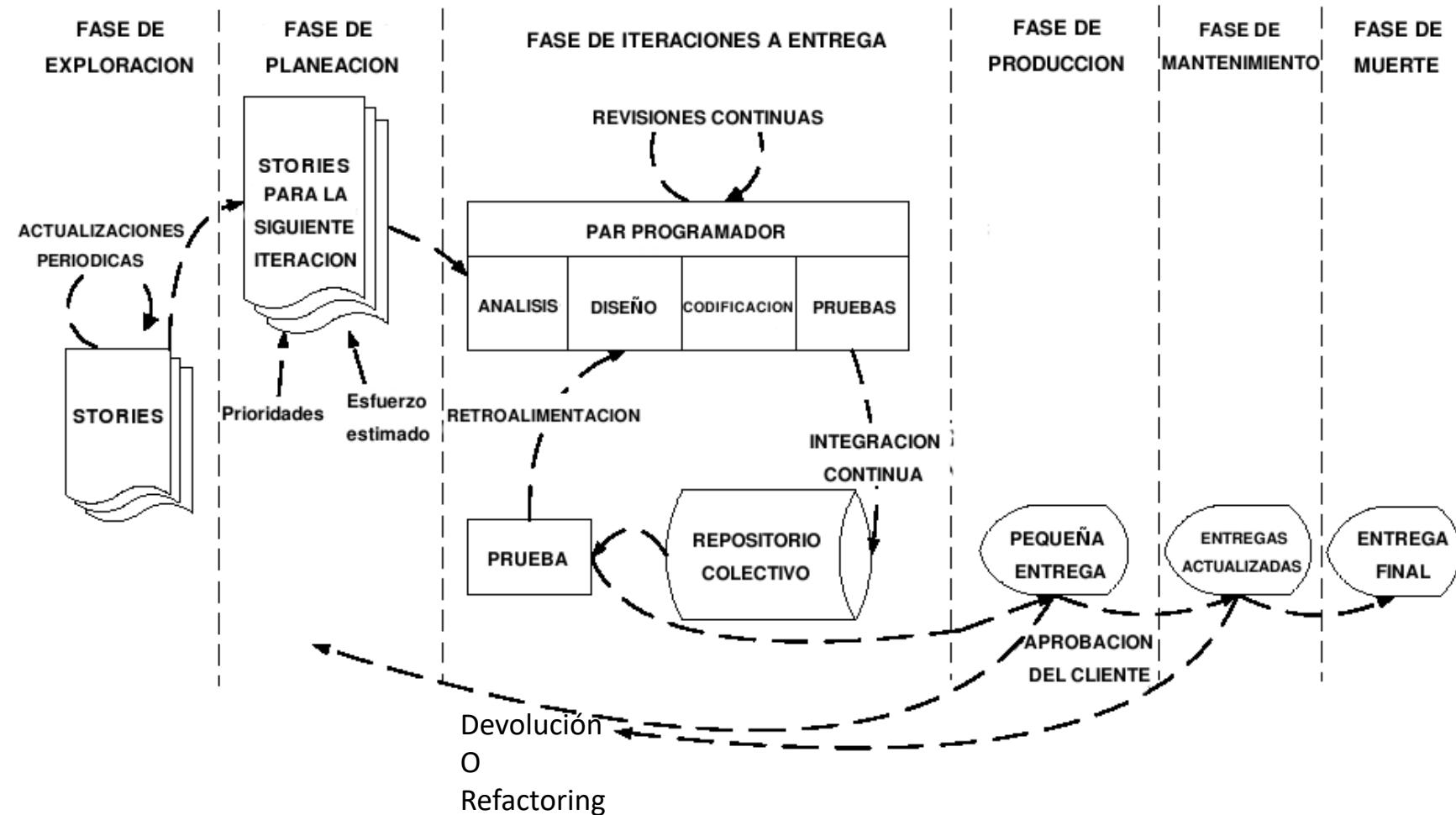
Pruebas: Nunca se puede estar seguro de algo hasta haberlo probado.

Escuchar: Escuchar los requisitos del cliente acerca del sistema a crear.

Diseño: Crear una estructura del diseño para evitar problemas.

Metodología Ágil XP

Fases de la metodología eXtreme Programming propuesta por Kent Beck



Metodología Ágil XP

XP Extreme Programming

Se fundamenta en 12 prácticas

El proceso de planificación

Define características que se incluirán en la versión siguiente combinando prioridades de negocio definidas con el cliente con estimaciones técnicas de los programadores.

Los pequeños “releases”

Un sistema simple en producción temprana, se actualiza frecuentemente en un ciclo corto

Metáfora

El sistema se define como una metáfora o conjunto de metáforas una historia compartida por clientes, managers y programadores que orienta todo el sistema, describiendo como funciona.

Diseño simple

Diseñar la solución más simple susceptible de implementarse en el momento. No implementar nada que no se necesite ahora. (Nivel de complejidad)

Recodificación

El diseño se mejora en todo el ciclo, manteniendo el software limpio, sin duplicación, simple y completo. Si funciona bien arréglelo de todos modos.

Programación en pares

La programación se hace entre dos personas (Nivelación técnica)

PLANIFICACION

DISEÑO

CODIFICACION

PRUEBAS

- 1. El juego de la planificación
- 2. Entregas pequeñas
- 3. Metáfora
- 4. Diseño simple
- 5. Recodificación
- 6. Programación en parejas
- 7. Propiedad colectiva
- 8. Integración continua
- 9. Semana de 40 horas
- 10. Cliente in situ
- 11. Estándares de programación
- 12. Pruebas

Metodología Ágil XP

XP Extreme Programming

Propiedad colectiva

Todo el código pertenece al grupo. Cualquiera puede cambiar cualquier parte del código en cualquier momento, siempre que escriba antes la prueba correspondiente.

Integración continua

Cada pieza se integra a la base de código apenas está lista, varias veces al día. Debe correrse la prueba antes y después de la integración. Hay un repositorio (solamente) dedicada a este propósito.

40 horas semana

Se evitan los sobretiempos excesivos y los equipos cansados. Evitar los "héroes" y eliminar el "proceso neurótico" (Salario emocional, amor a la compañía)

Cliente en sitio

El cliente debe estar presente y disponible a tiempo completo para el equipo. (Línea directa)

Estándar de codificación

Para compartir el código la codificación debe ser similar

Prueba

Desarrollo orientado a las pruebas UT, TDD.

PLANIFICACION

DISEÑO

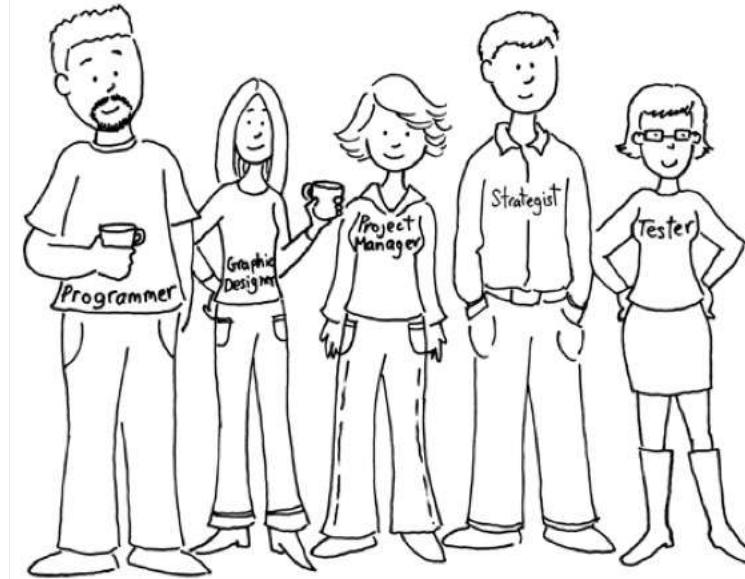
CODIFICACION

PRUEBAS

- 1. El juego de la planificación
- 2. Entregas pequeñas
- 3. Metáfora
- 4. Diseño simple
- 5. Recodificación
- 6. Programación en parejas
- 7. Propiedad colectiva
- 8. Integración continua
- 9. Semana de 40 horas
- 10. Cliente in situ
- 11. Estándares de programación
- 12. Pruebas

Metodología Ágil XP

Roles en XP



Prácticas ágiles en XP

Plan de entregas “Release Plan”

Es donde se definen las iteraciones que tendrá el proyecto, las HU a desarrollar por iteración y se acuerdan las entregas a realizar

Iteración	Historia de Usuario	Puntos de Historia	Fecha (inicio: Marzo 29 de 2023)	Par Programador
Iteración 1	HU1	3	**	Juan-Maria Dani-Pedro
	HU2	4	7	
Iteración 2	HU3	2	**	Juan-Maria Sofia-Juliana Dani-Pedro
	HU4	4	**	
	HU6	2	8	
Iteración 3	HU5	4	**	Sofia-Juliana Juan-Maria
	HU7	3	7	

Nota: Si una HU es muy grande (EJ: 6 puntos de historia), sería más adecuado dividirla en dos historias (EJ: una de 4 y otra de 2) y distribuirlas en las iteraciones.

La velocidad de desarrollo debería ser similar para cada iteración.

** : Se calcula a partir de la fecha de inicio y depende de la experiencia del equipo de desarrollo (horas por punto)

Nota: El plan de entregas puede cambiar en cada iteración

Prácticas ágiles en XP

Velocidad del proyecto:

La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto y se mide en puntos.

Para estimarla la velocidad de un proyecto se requiere sumar el número de puntos de cada historia de usuario que se determina implementar en una iteración.

Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración.

Prácticas ágiles en XP

Programación en pareja:

La metodología XP aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado.

El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método es adecuado y si está bien diseñado.

ROLES



Navegador /Copiloto

Programador encargado de supervisar y debatir entorno al código que se está escribiendo.

Controlador / Piloto

Programador encargado de escribir el código.

Intercambio Regular entre pares.

VENTAJAS

- I. Los estudios han demostrado que después de entrenar para las "**habilidades sociales**" implicadas, parejas de programadores son más de **dos veces productivos** que uno para una tarea dada. Según *The Economist*.

DESVENTAJAS

- II. "Laurie Williams de la universidad de Utah en Salt Lake City ha demostrado que los programadores emparejados son solamente **15% más lentos** de dos programadores trabajando independientemente, **pero producen 15% menos errores**. Y ya que la prueba y depuración son a menudo muchas veces más costosa que la programación inicial, esto es un **resultado impresionante**"
 - ✓ Moral Mejorada.
 - ✓ Propiedad colectiva del código.
 - ✓ Método más eficaz de capacitación.
 - ❖ Aumento de los costes.

Prácticas ágiles en XP

Reuniones diarias.

Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.



- 15 Minutos
- De pie
- Sin café

Prácticas ágiles en XP

..... **Diseños simples:** La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implemémentable que a la larga costará menos tiempo y esfuerzo desarrollar.

..... **Glosarios de términos:** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

..... **Riesgos:** Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema (**spyke**).

Prácticas ágiles en XP

Funcionalidad extra

Nunca se debe añadir funcionalidad extra al sistema aunque se piense que en un futuro será utilizada. Típicamente sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

De las funcionalidades desarrolladas:

- 7% se usan “siempre”
- 13% se usan “a menudo”
- 16% “a veces”
- 19% “pocas veces”

45% “NUNCA”

Fuente: Standish Group

Refactorizar

Refactorizar es mejorar y modificar la estructura y codificación (código) ya creado sin alterar su funcionalidad. Refactorizar supone revisar de nuevo el código para procurar optimizar su funcionamiento. Es muy común rehusar código ya creado que contienen funcionalidades probadas.

Prácticas ágiles en XP

Codificación.

.....La codificación debe hacerse ateniendo estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente para facilitar su comprensión y escalabilidad.

<https://django-best-practices.readthedocs.io/en/latest/code.html>

<https://www.geeksforgeeks.org/best-practices-for-professional-developer-django-framework/>

.....La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

<https://docs.djangoproject.com/en/4.1/topics/performance/>

Prácticas ágiles en XP

Pruebas

Hacer pruebas valida el funcionamiento de la aplicación.

Pruebas Unitarias : es un método que prueba una unidad estructural de código.

Pruebas de aceptación: Se especifica por medio de criterios de aceptación.



1

Pruebas | CRITERIOS DE ACEPTACIÓN

Técnica: Acción Reacción



2

Pruebas | CRITERIOS DE ACEPTACIÓN

Técnica: Escenarios

Criterios de aceptación

Características de los Criterios de Aceptación

Para medir la calidad de un criterio de aceptación se utiliza el método SMART en el que se han de cumplir en lo máximo posible los siguientes criterios:

- S** - Specific (Específicas)
- M** - Measurable (Medibles)
- A** - Achievable (Alcanzables)
- R** - Relevant (Relevantes)
- T** - Time-boxed (Limitados en el tiempo)

Criterios de aceptación

Pruebas: Técnica acción/reacción

Son las condiciones que una historia de usuario debe satisfacer para ser aceptada por un usuario, cliente o stakeholder.

Si todas las HU son aceptadas el producto de software es aceptado.

Se presentan como un conjunto de sentencias redactadas de tal manera que conduzcan a una respuesta clara de “aceptado/rechazado”

Un criterio de Aceptación debe ser escrito en formato de Pruebas de Aceptación.

Prácticas ágiles en XP

Pruebas: Técnica acción reacción

Dada una condición, cuando ocurre un evento o acción, entonces sucederá una consecuencia.
Criterio de aceptación expresado en términos funcionales (Acción/Reacción) de uso del sistema.



Item	Criterio de aceptación
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

Acción

Reacción

Criterios de aceptación

Técnica de escenarios:

- Restringir la forma en que el usuario procedería y el sistema correspondería,
- Elimina toda la información innecesaria.
- Tiene la ventaja de que permite al equipo de desarrollo ir dando por hechas las funcionalidades y los casos que de ella se origina mientras las van implementando.
- Un test de aceptación puede comprobar más de un criterio de aceptación.
- Permiten usar más de un usuario para describir el escenario del criterio.
- Su mayor beneficio es que los criterios de aceptación pueden ser trasladados directamente de la historia de usuario a un test de aceptación automático.

Criterios de aceptación: técnica de escenarios

Usa la sintaxis de **Gherkin** creada específicamente para las descripciones de comportamiento de un software.

La sintaxis de gherkin es la siguiente:

```
*  
1 Scenario: Some determinable business situation  
2   Given some precondition  
3     And some other precondition  
4     When some action by the actor  
5       And some other action  
6       And yet another action  
7     Then some testable outcome is achieved  
8       And something else we can check happens too
```



Criterios de aceptación: técnica de escenarios

(Scenario) Escenario [Número de escenario] [Título del escenario]:

(Given) Dado que [Contexto] y adicionalmente [Contexto]

(When) cuando [Evento] y [evento]

(Then) entonces el sistema [Resultado / Comportamiento esperado]

Número de escenario: Número (ejemplo 1, 2 o 3 ó CA1_HU1, CA2_HU1), que identifica al escenario asociado a la historia.

Título del escenario: Describe el contexto del escenario que define un comportamiento.

Contexto: Proporciona mayor descripción sobre las condiciones que desencadenan el escenario.

Evento: Representa la acción que el usuario ejecuta, en el contexto definido para el escenario.

Resultado / Comportamiento esperado: Dado el contexto y la acción ejecutada por el usuario, la consecuencia es el comportamiento del sistema en esa situación.

Criterios de aceptación: técnica de escenarios

Ejemplo: HU: Como cliente del banco deseo retirar dinero de un cajero automático para evitar ir al banco.

Scenario 1: La cuenta tiene saldo disponible
Given la cuenta tiene saldo disponible
and que la tarjeta es válida
and que el cajero tiene dinero disponible
When el cliente solicita dinero
Then la cuenta es debitada
and el dinero es entregado al cliente
and el cliente recupera su tarjeta

Escenario 1: La cuenta tiene saldo disponible
Dado que la cuenta tiene saldo disponible
y que la tarjeta es válida
y que el cajero tiene dinero disponible
Cuando el cliente solicita dinero
Entonces la cuenta es debitada
y el dinero es entregado al cliente
y el cliente recupera su tarjeta

Criterios de aceptación: técnica de escenarios

Ejemplo: HU: Como cliente del banco deseo retirar dinero de un cajero automático para evitar ir al banco.

Scenario 2: El retiro excede el saldo disponible en el banco

Given el retiro excede el saldo disponible en el banco

and que la tarjeta es válida

and que el cajero tiene dinero disponible

When el cliente solicita dinero

Then el cajero muestra un mensaje negando el retiro

and el dinero no es entregado al cliente

and el cliente recupera su tarjeta

Criterios de aceptación: técnica de escenarios

Ejemplo: HU: Como operador de oficina quiero ingresar al sistema para trabajar en mis actividades

Escenario 1: El usuario operador de oficina desea ingresar al sistema
Dado que el usuario operador tiene una cuenta habilitada en el sistema
y su login es prueba
y su password es tester1
Cuando el usuario operador escribe su login
y el usuario operador escribe su contraseña
Entonces el sistema le muestra un mensaje que dice “Login exitoso”
y lo redirige a la pantalla de inicial para el rol de operador

<https://cucumber.io/docs/guides/10-minute-tutorial/>



Mensaje final sobre metodologías

Ninguna metodología hará el trabajo por ti,
porque ninguna metodología trabaja sola.



Preguntas ?

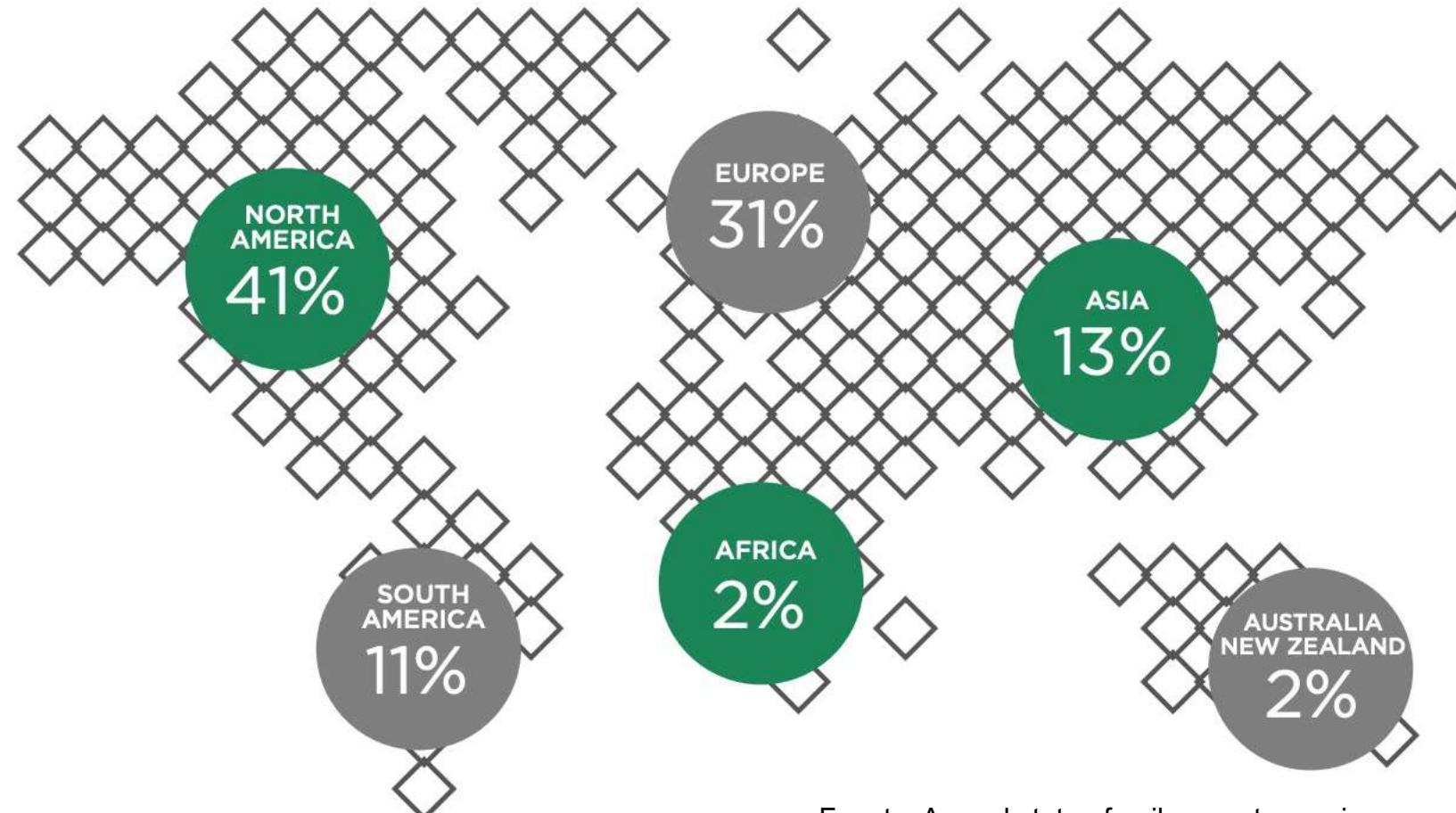




Uso de las metodologías ágiles Estadísticas

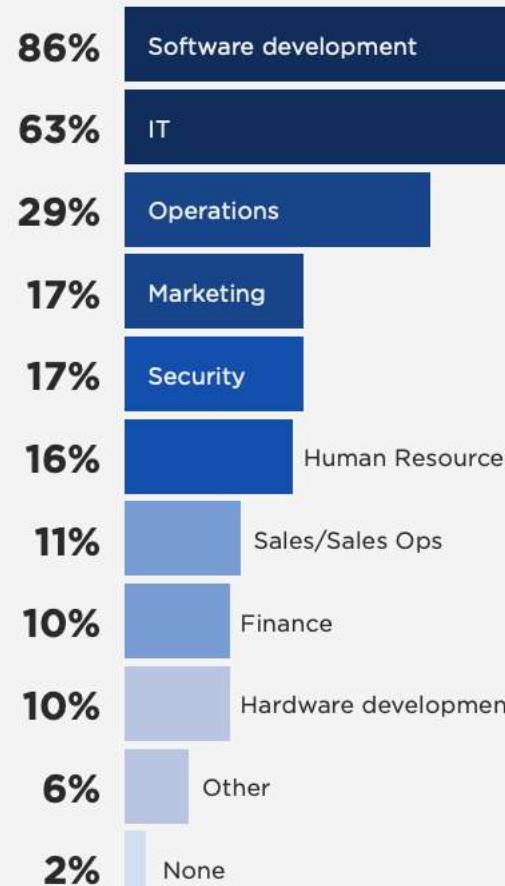


Uso de las metodologías ágiles



Uso de las metodologías ágiles

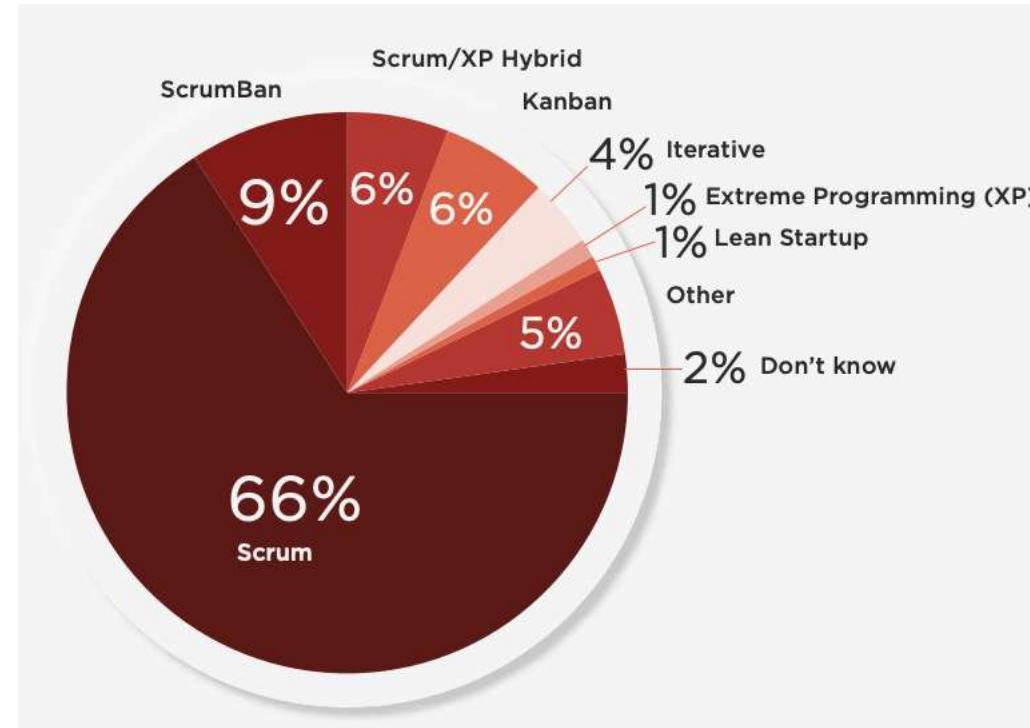
Respuestas de empresas que trabajan en:



Fuente: Annual state of agile report - versionone

Uso de las metodologías ágiles

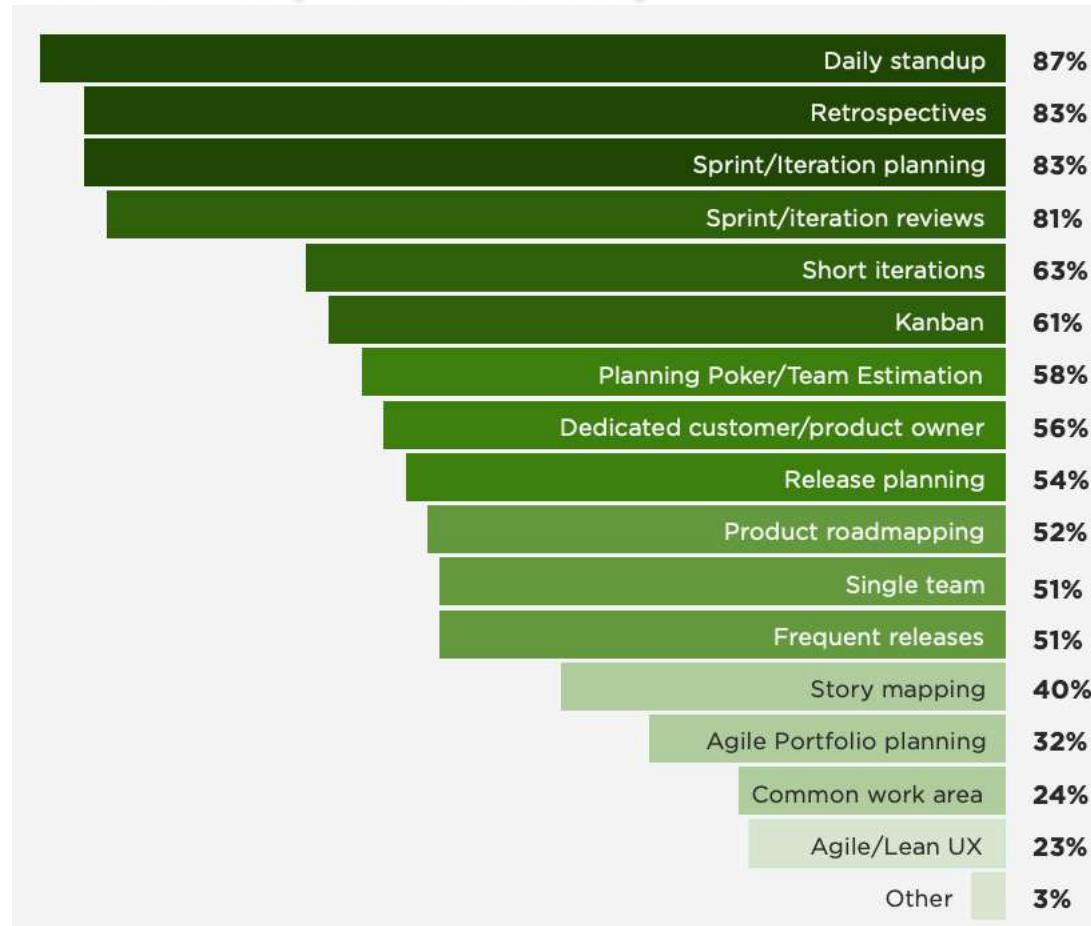
Metodologías más usadas



Fuente: Annual state of agile report - versionone

Uso de las metodologías ágiles

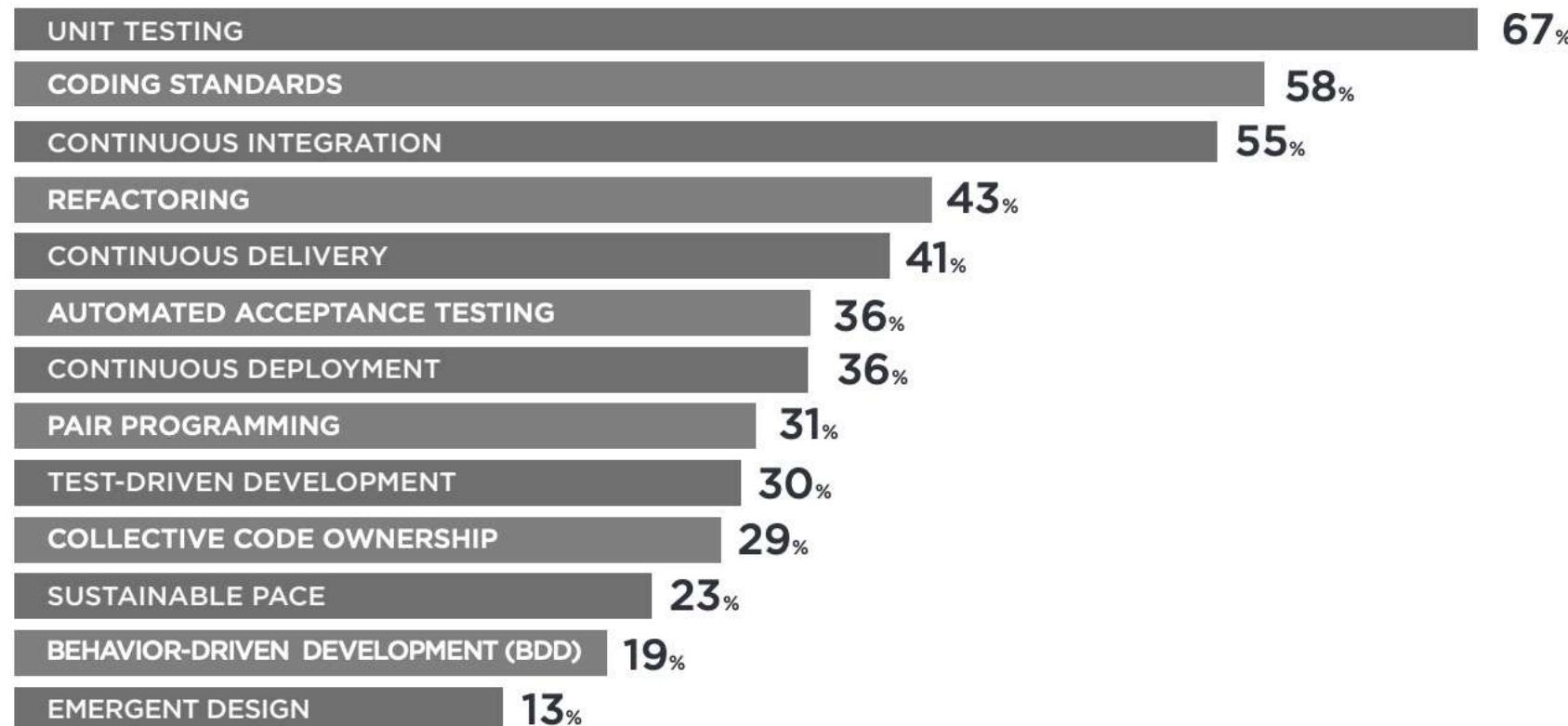
Prácticas ágiles colaborativas (las más usadas)



Fuente: Annual state of agile report - versionone

Uso de las metodologías ágiles

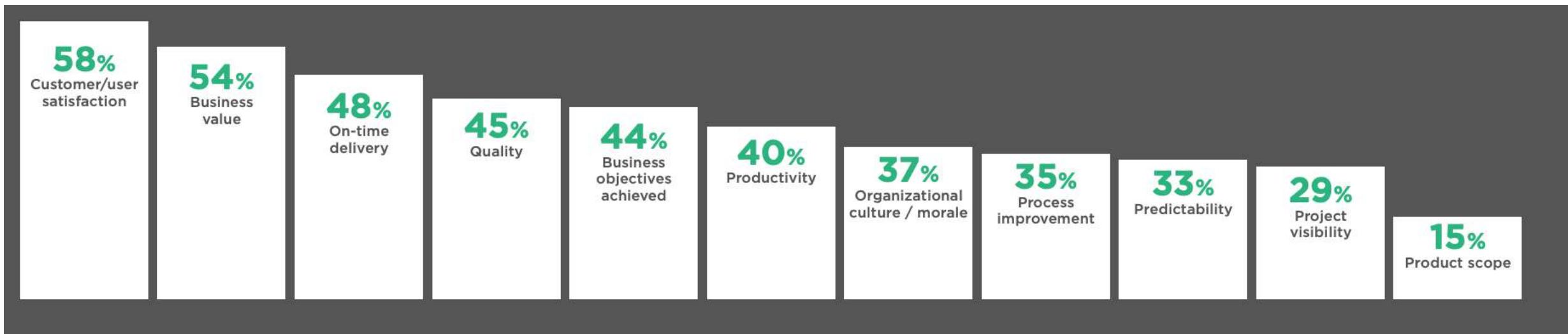
Prácticas ágiles técnicas



Fuente: Annual state of agile report - versionone

Uso de las metodologías ágiles

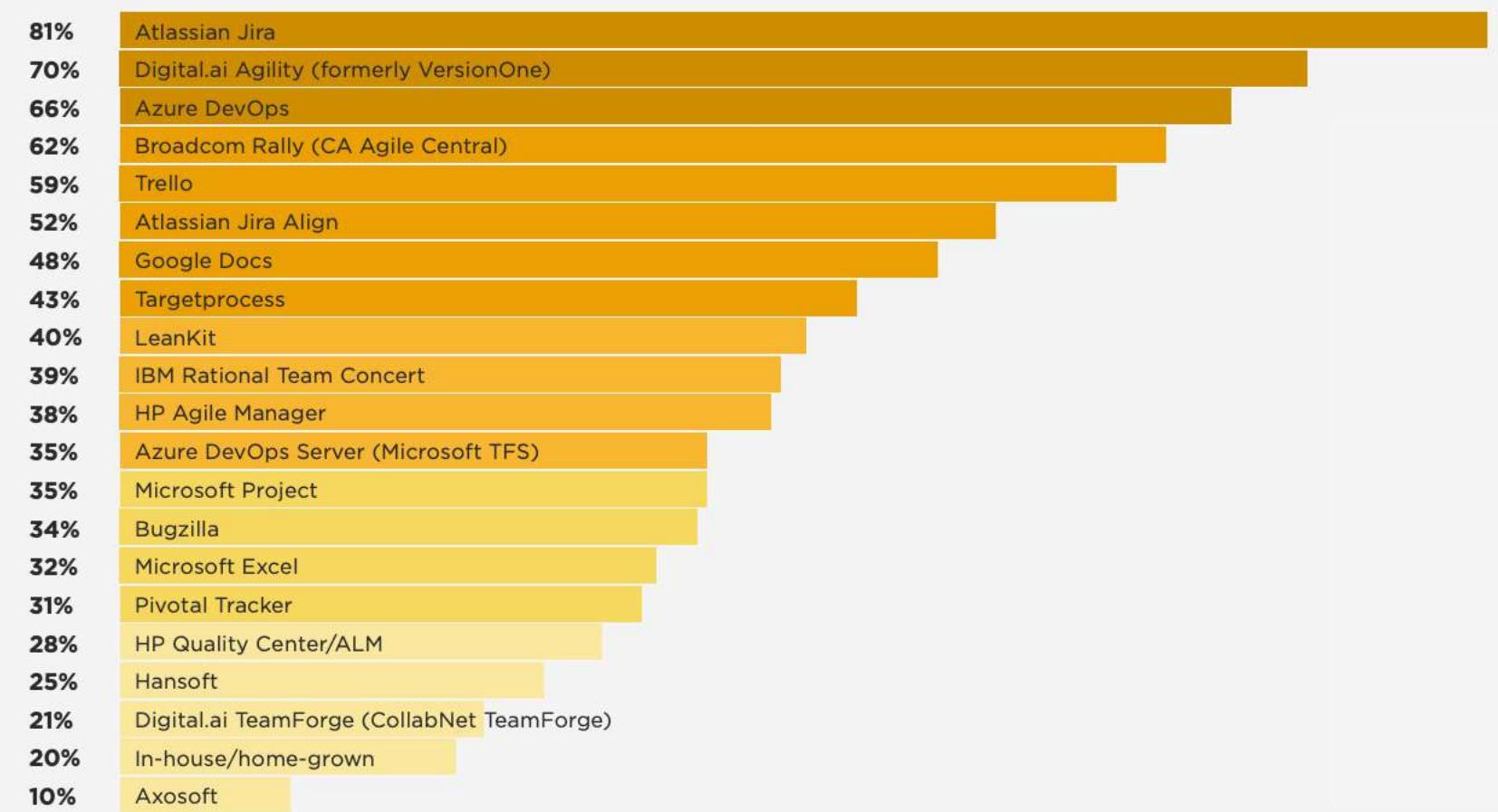
Medición del éxito de uso de las prácticas ágiles



Fuente: Annual state of agile report - versionone

Uso de las metodologías ágiles

Herramientas usadas



Fuente: Annual state of agile report - versionone

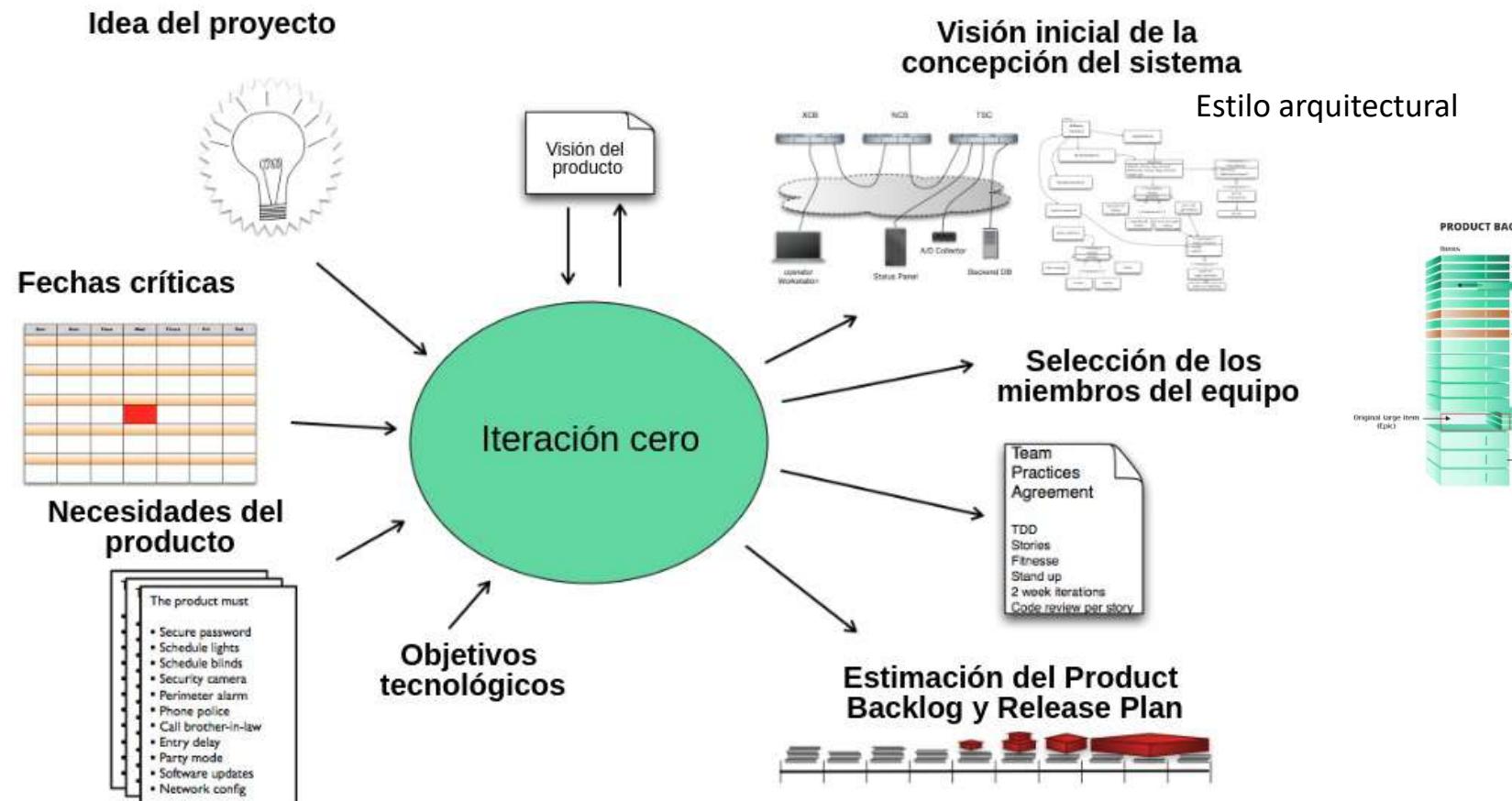
¿Cómo iniciar el desarrollo de proyecto de software usando prácticas ágiles?



Iniciando un proyecto de software usando prácticas ágiles

El Sprint cero (Iteración Cero)

En algunos equipos es frecuente el uso del llamado Sprint cero, **cuyo objetivo son los preparativos previos a comenzar el desarrollo**. Así, normalmente, durante el Sprint 0 se realizan diferentes tareas.



Iniciando un proyecto de software usando prácticas ágiles

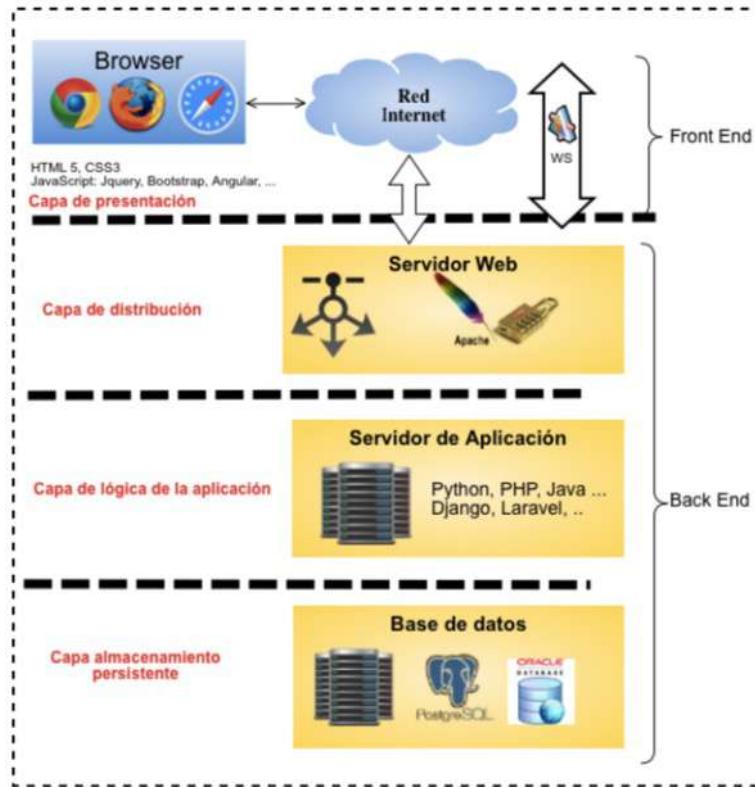
Principales actividades en el Sprint cero:

- Idea inicial del sistema a desarrollar.
- Fechas críticas.
- Necesidades (requerimientos a alto nivel).
- Restricciones del proyecto.
- Selección del equipo de desarrollo.
- Se selecciona las tecnologías y se dejan listos los entornos de desarrollo.
- Se acuerda la metodología de desarrollo a usar.
- Hacer la elicitación del sistema (Product Backlog)
- Se trabaja en el product backlog, principalmente en dejar listas las historias de usuario, priorizadas y estimadas.
- Se selecciona el Estilo arquitectural, Modelo de datos y Modelo de despliegue.
- Release plan (Se hace una distribución de historias de usuario por Sprints).
- Se define como se hará el diseño visual del sistema (Plantilla)
- ...

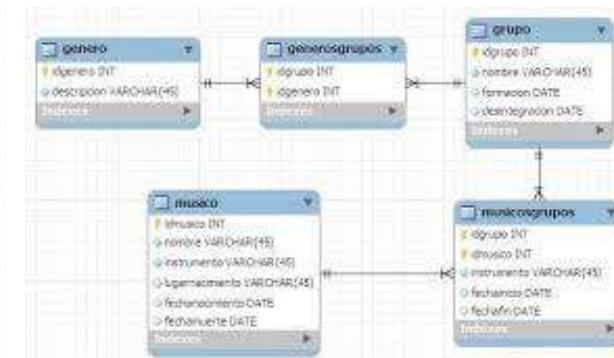
Iniciando un proyecto de software usando prácticas ágiles

Modelos arquitecturales del sistema

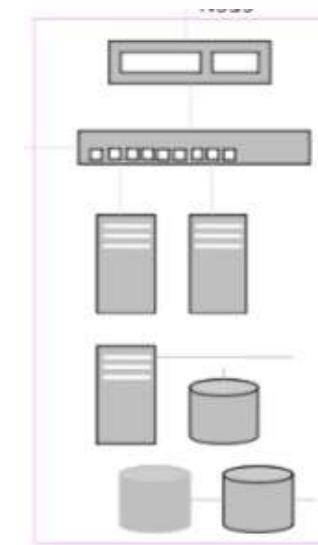
“Simple Sketches for Diagramming your Software Architecture “



Estilo arquitectural de la aplicación



Modelo de datos (inicial)



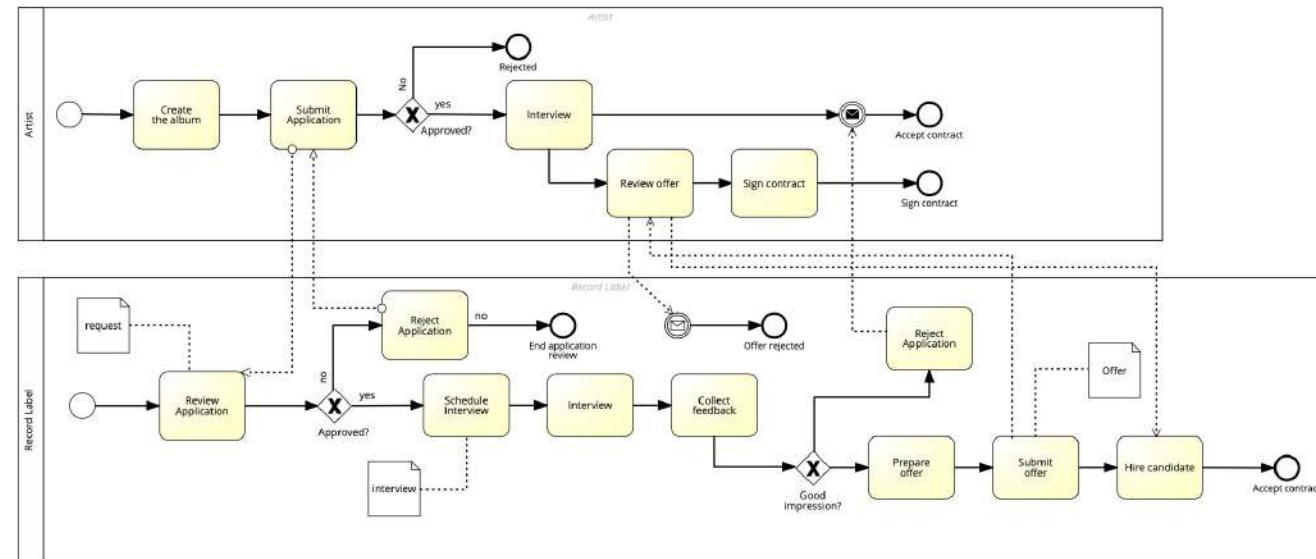
Modelo de despliegue



Preguntas ?



Introducción a BPMN (Business Process Model and Notation)

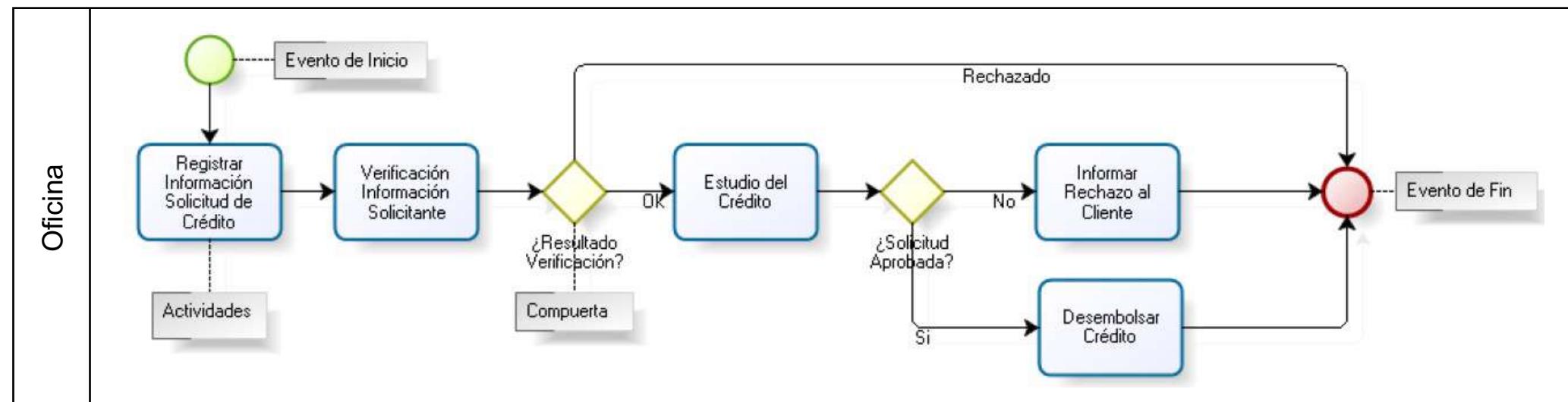




Introducción a BPMN (Business Process Model and Notation)

- Es un estándar para el modelamiento de procesos de negocio.
- Ayuda a entender los procesos de negocio desde una notación gráfica.

Proceso solicitud de crédito



Tomado de BPMN Business Process Modeling Notation – Bizagi Process Modeler



Introducción a BPMN (Business Process Model and Notation)

Las historias de usuario desde los procesos de negocio

- Usada como una técnica en el proceso de elicitation de requisitos.
- Los requisitos se obtienen desde la perspectiva de los usuarios mediante el modelado de procesos de negocio.
- *Los desarrolladores de software han reconocido la importancia de entender los procesos de negocio en la elicitation de requisitos.*



Introducción a BPMN (Business Process Model and Notation)

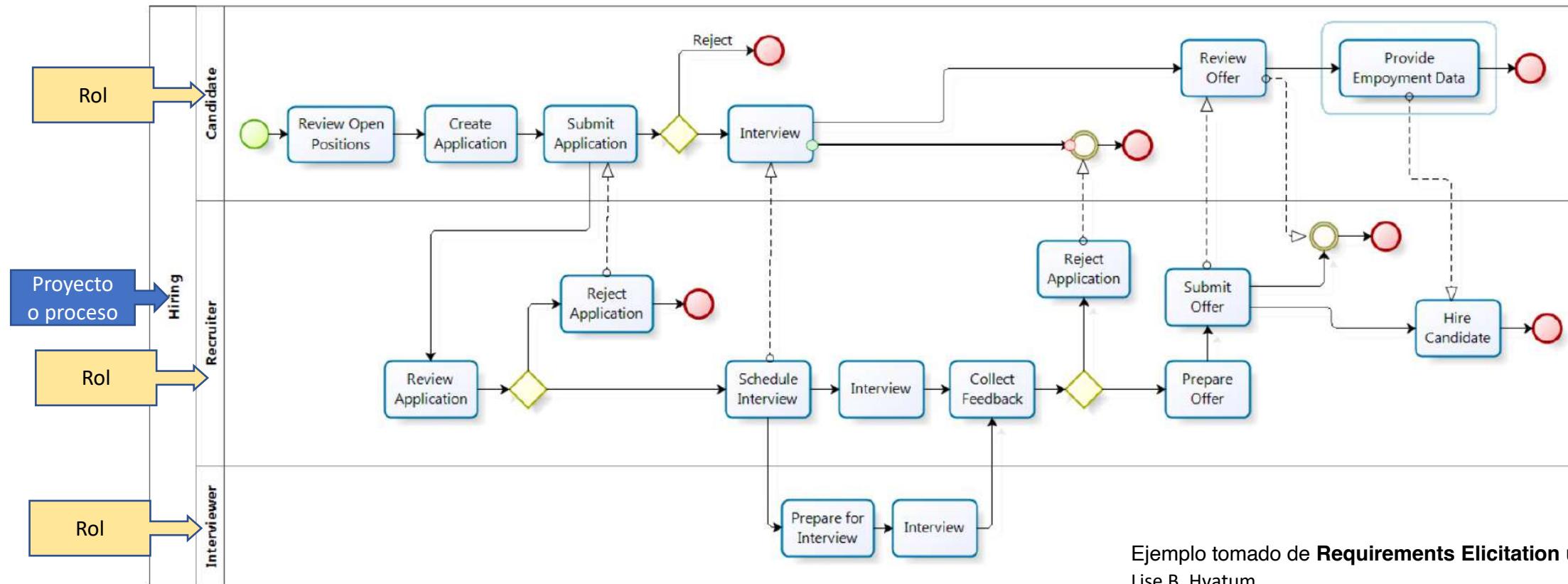
Las historias de usuario desde los procesos de negocio – Pasos

1. Modelar los procesos de negocio utilizando BPM (notación BPMN)
2. Analizar los elementos del proceso para extraer historias de usuario para cada rol
3. Plantear las historias de usuario
4. Realizar un análisis adicional del proceso de negocio, en conjunto con usuarios/clientes, para generar nuevas historias de usuario.

Introducción a BPMN (Business Process Model and Notation)

Las historias de usuario desde los procesos de negocio – Ejemplo

Paso 1 y Paso 2



Ejemplo tomado de Requirements Elicitation using BPM
Lise B. Hvatum

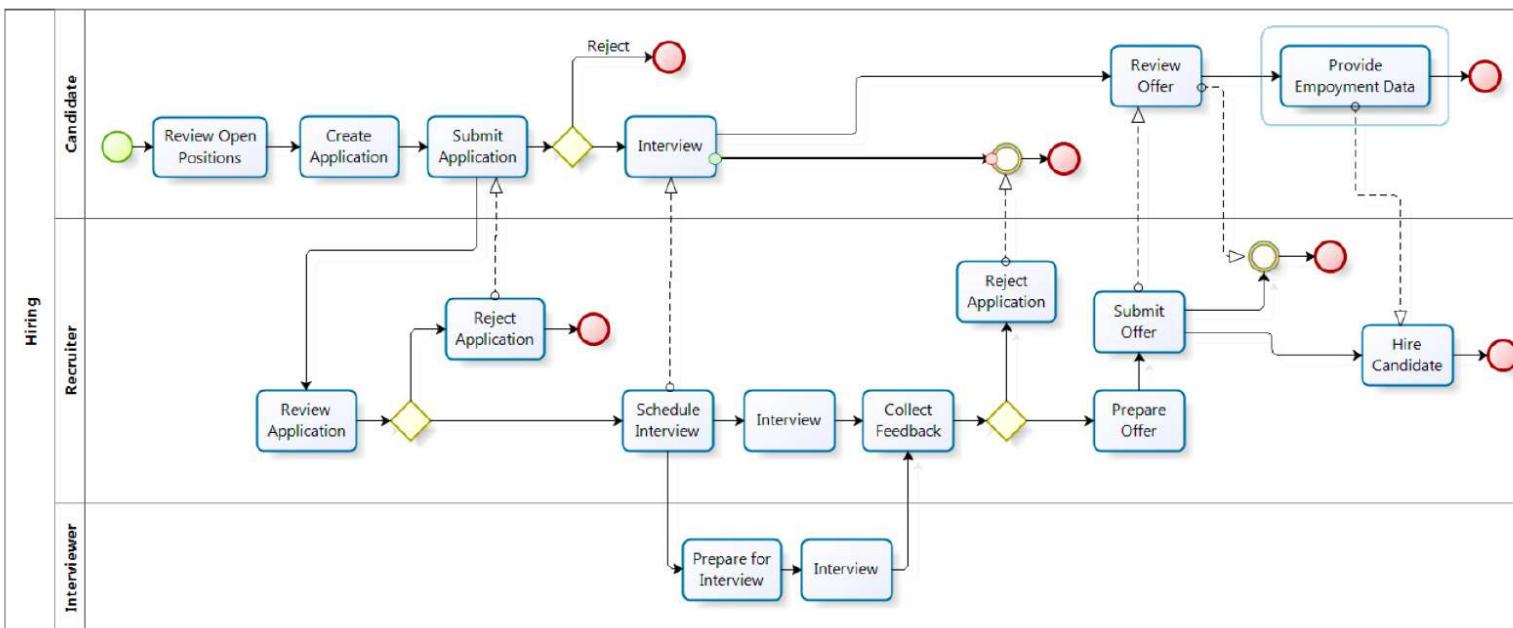


Introducción a BPMN (Business Process Model and Notation)

Las historias de usuario desde los procesos de negocio – Ejemplo

Paso 1 y Paso 2

Paso 3



- As a Candidate I can see open positions
- As a Candidate I can submit an application
- As a Candidate I get notified of a rejected application
- As a Candidate I can receive an invitation for an interview
- As a Candidate I can receive an offer for a position
- As a Candidate I can accept the offer for a position
- As a Candidate I can reject the offer for a position
- As a Candidate I can provide employment information



Introducción a BPMN (Business Process Model and Notation)

Desafíos de esta técnica

- Decidir el nivel de detalle en que se modelará el proceso de negocio
- Cómo manejar las variaciones en el proceso de negocio (secuencias de las actividades)



Introducción a BPMN (Business Process Model and Notation)

¿Cómo modelar un proceso de negocio?



Introducción a BPMN (Business Process Model and Notation)

Elementos básicos BPMN

Objetos de flujo (Flow Objects)

Objetos de Conexión (Connecting Objects)

Canales - Carriles (Swimlanes)

Artefactos (Artifacts)



Introducción a BPMN (Business Process Model and Notation)

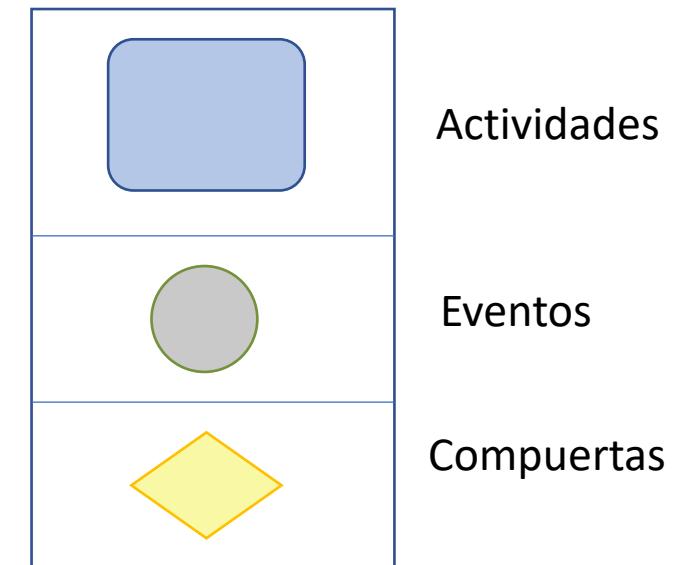
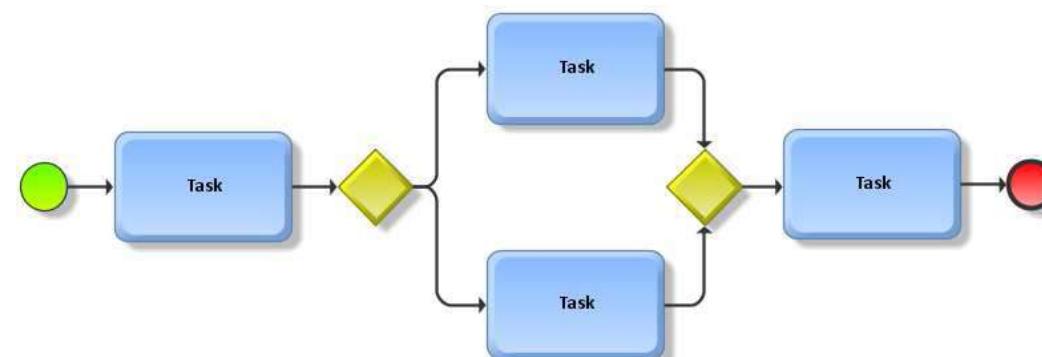
Objetos de Flujo

Principales elementos que ayudan a definir el comportamiento de los procesos.

Actividades – Activities

Eventos – Events

Compuertas - Gateways





Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo – Actividades

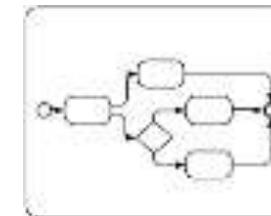
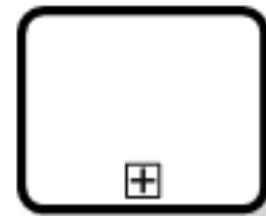
Representan trabajo o tareas realizadas por los participantes del proceso.

Pueden ser manuales o automáticas, pueden ser compuestas o no.

Tareas o Actividades (HU)



Subprocesos





Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo – Actividades

Tareas

Nombre	Notación	Descripción
Tarea		Indica un trabajo a realizar. Es atómica.
Tarea de enviar		Diseñada para enviar un mensaje. Se completa una vez el mensaje es enviado.
Tarea de recibir		Diseñada para recibir un mensaje.



Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo – Actividades

Tareas

Nombre	Notación	Descripción
Invocación de servicio		Utiliza algún tipo de servicio, éste puede ser un servicio web o una aplicación automática. No tiene intervención humana. Ej: Hacer interfaces con otros sistemas.
Tarea Manual		Es una tarea que no utiliza ningún motor de ejecución de procesos. Ej: archivar documentos.
Tarea de script		Se ejecuta automáticamente. No hay interacción humana. Ej: envío de correos electrónicos.



Introducción a BPMN (Business Process Model and Notation)

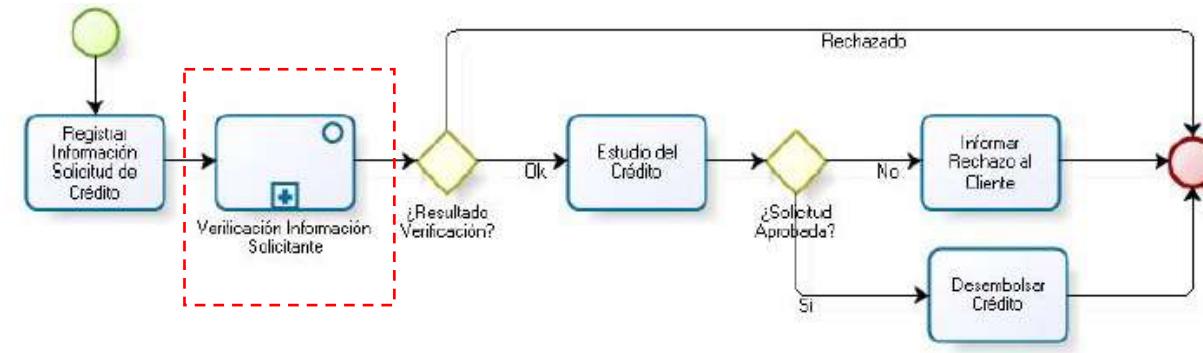
Objetos de Flujo – Actividades

Subprocesos

Nombre	Notación	Descripción
Subproceso		Es una actividad compuesta que está incluida dentro de un proceso.

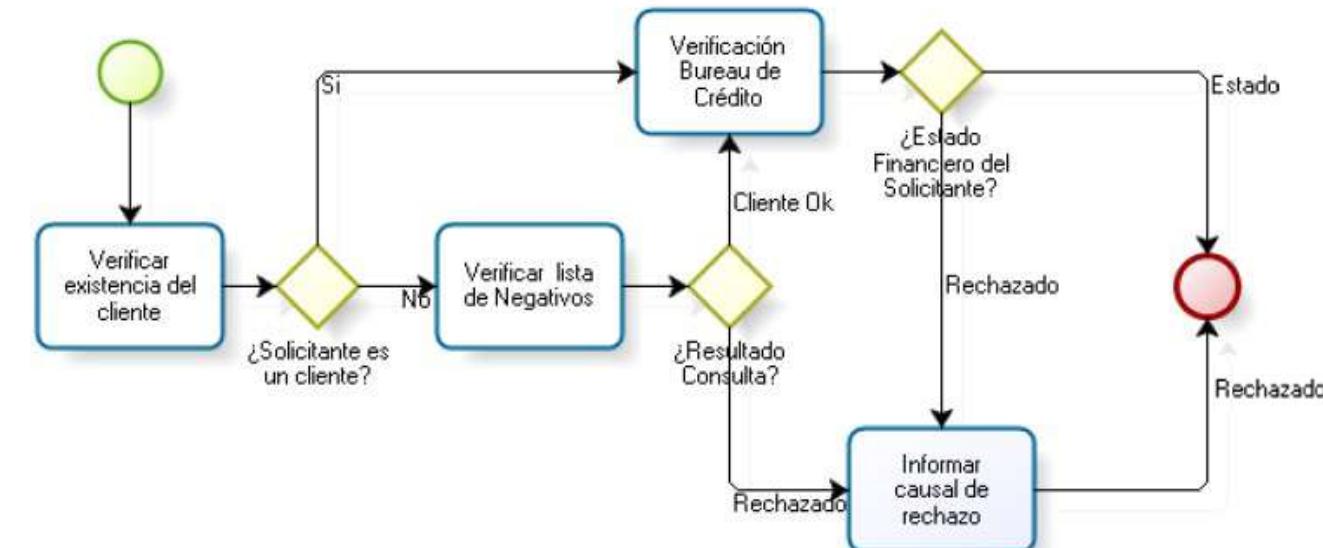
Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo – Actividades



Proceso de solicitud de un crédito

El subprocesso de verificación de Información del solicitante es el siguiente.

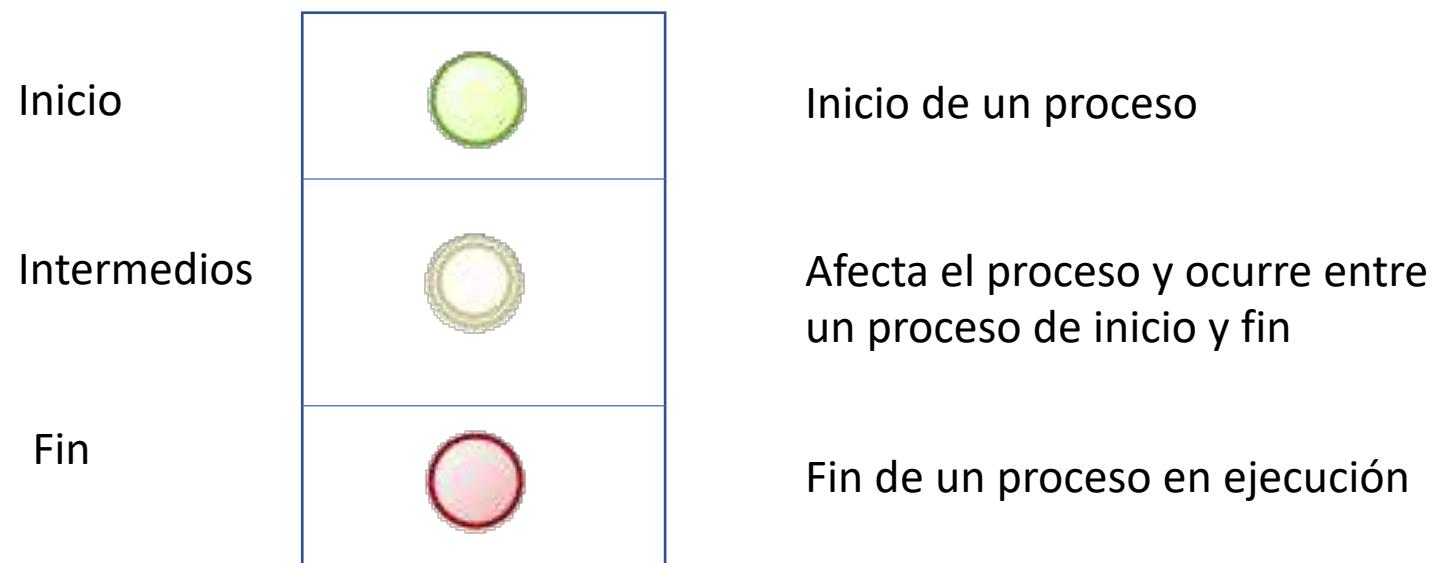




Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo – Eventos

Es algo que sucede durante el proceso. Normalmente tiene una causa que lo acciona.





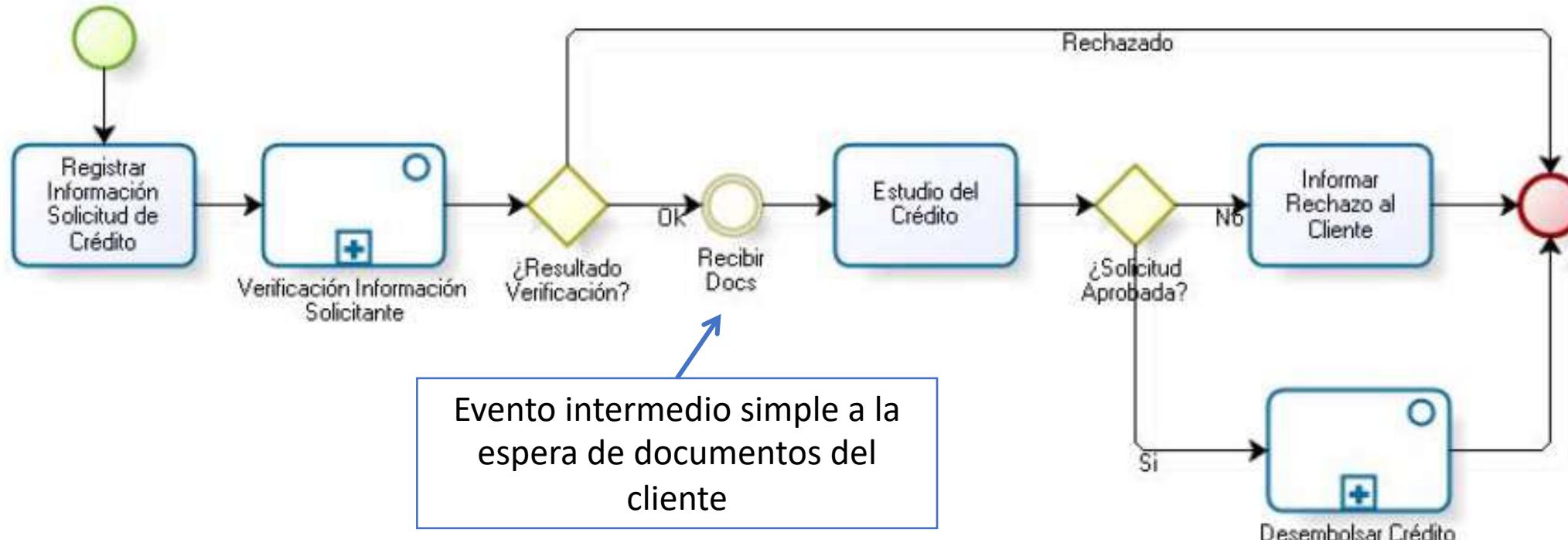
Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo - Eventos

EVENTO	Inicio	Intermedio	Fin	Descripción
Mensaje				<p>Recepción y envío de mensajes entre procesos. El de inicio. El proceso puede iniciar por el recibo de un mensaje de un participante (otro proceso). El intermedio puede usarse para enviar como para recibir mensajes. Puede lanzarse así como atraparse. En el caso del fin se envía un mensaje al concluir con el proceso.</p>
Tiempo				<p>El inicio del proceso se activa cuando se sucede una fecha u hora. El intermedio se utiliza como mecanismo de retraso. Como indicar un tiempo de espera de una actividad.</p>



Introducción a BPMN (Business Process Model and Notation)



Tomado de BPMN Business Process Modeling Notation – Bizagi Process Modeler



Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo - Compuertas

Usadas para controlar la divergencia y convergencia del flujo.

Tipo de decisión	Notación	Descripción
Decisión exclusiva		Selecciona solo un flujo de secuencia de las posibles alternativas existentes. Basada en datos del sistema.
Decisión paralela		Todos los flujos de secuencia son activados en el caso de la bifurcación. En la convergencia se espera que todos los flujos de secuencia activados se completen para continuar.
Decisión inclusiva		Inclusiva o multidecisión. Al menos un flujo de secuencia es activado cuando se presenta la bifurcación. Cuando ocurre la convergencia espera que todos los flujos que fueron activados lleguen para activar el saliente.



Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo - Compuertas

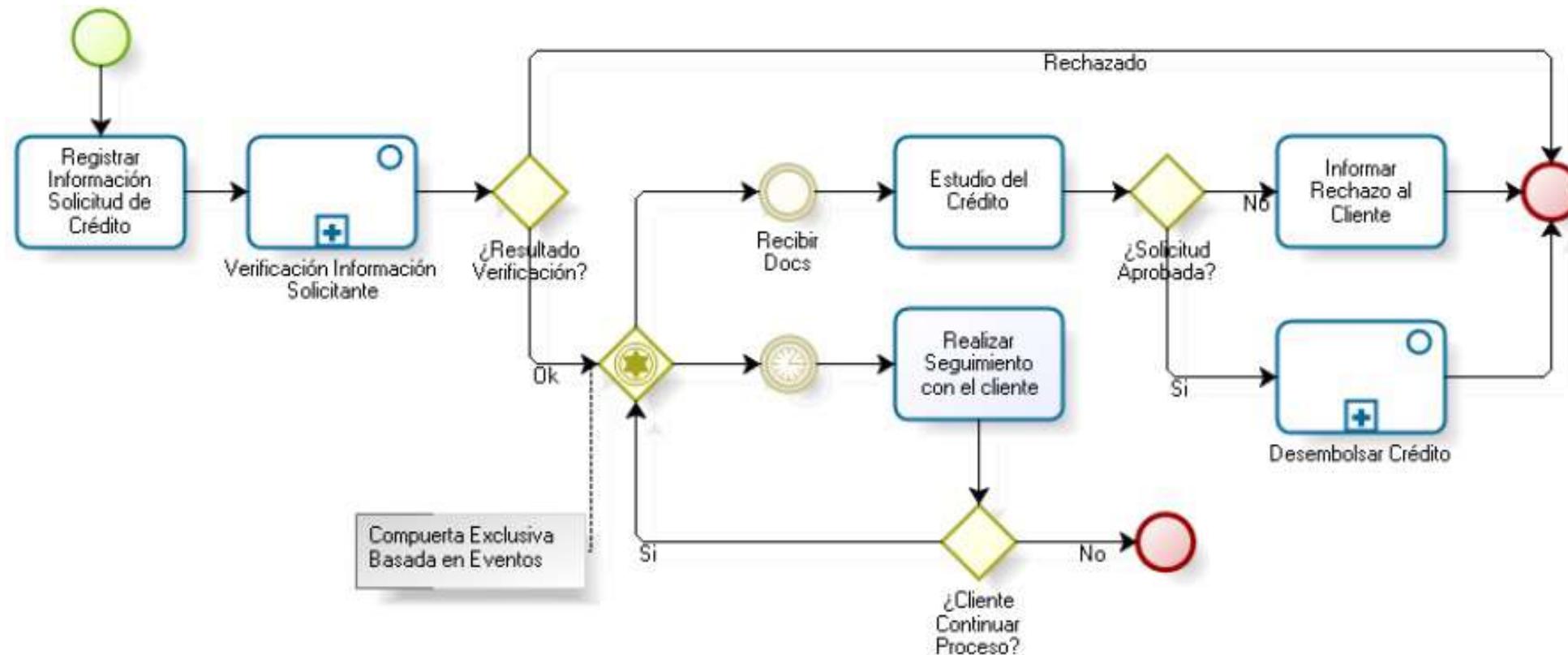
Usadas para controlar la divergencia y convergencia del flujo.

Tipo de decisión	Notación	Descripción
<u>Decisión basada en evento</u>		Se activará un flujo saliente dependiendo del evento que ocurra primero. Se basa en eventos o mensajes.
<u>Decisión compleja</u>		Se utiliza cuando se presenta la situación de tener que decidir entre 3 flujos de 5.



Introducción a BPMN (Business Process Model and Notation)

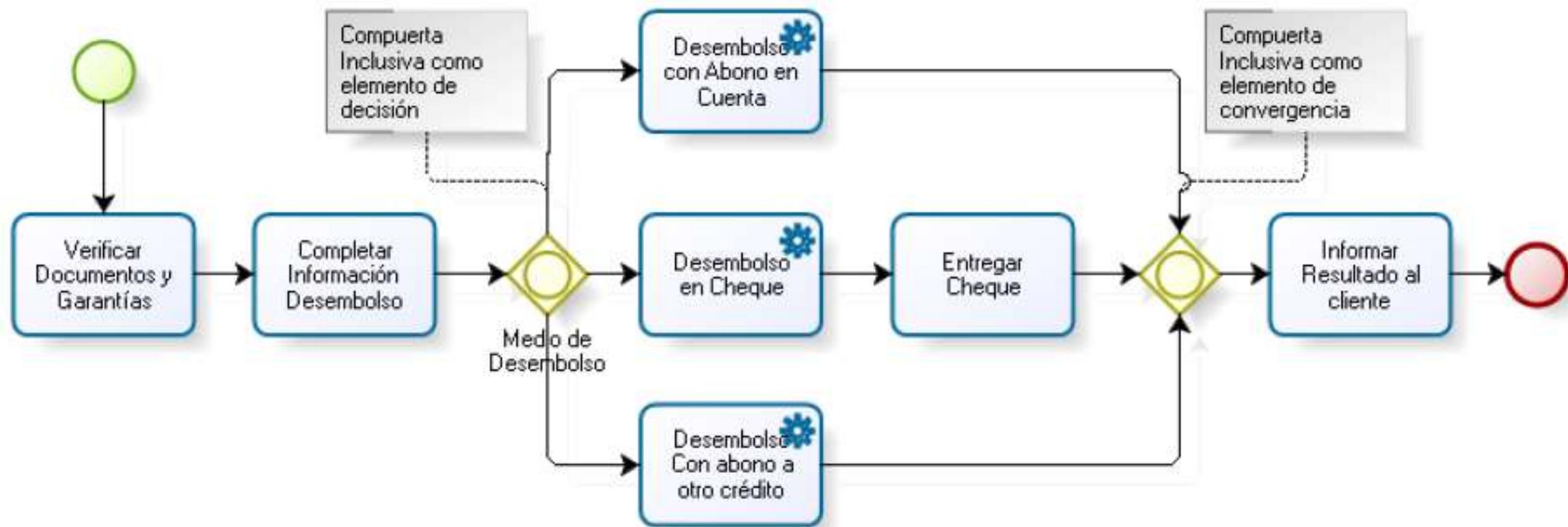
Objetos de Flujo - Compuertas





Introducción a BPMN (Business Process Model and Notation)

Objetos de Flujo - Compuertas





Introducción a BPMN (Business Process Model and Notation)

Conectores

Son los que conectan los objetos de flujo. Se clasifican en flujos de secuencia, de mensaje y de asociación.

Flujo de secuencia

- Sequence Flow



Flujo de mensaje

- Message Flow



Asociación

- Association



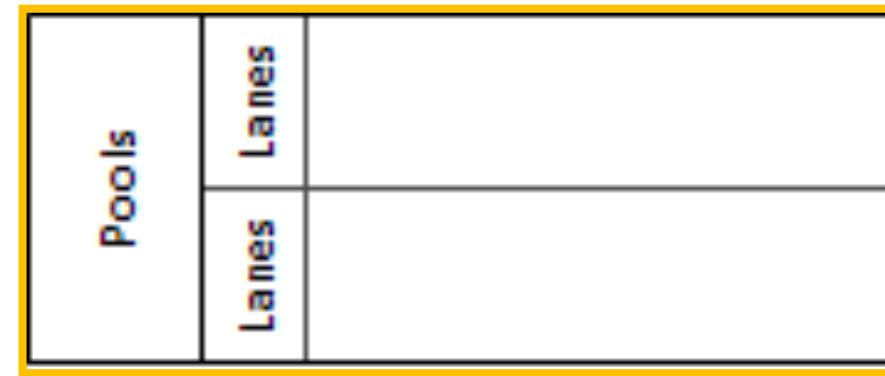


Introducción a BPMN (Business Process Model and Notation)

Canales

Utilizados para organizar las actividades del flujo. Son contenedores gráficos de las actividades de un proceso. Pueden ser áreas funcionales, sistemas, procesos, personas.

- Pool
- Lane





Introducción a BPMN (Business Process Model and Notation)

Artefactos

Se encargan de proveer información adicional sobre el proceso. No afectan la secuencia del flujo.
Ejemplo: una carta, un correo electrónico, un documento, un mensaje

Objetos de datos



[Descripción]

Grupo

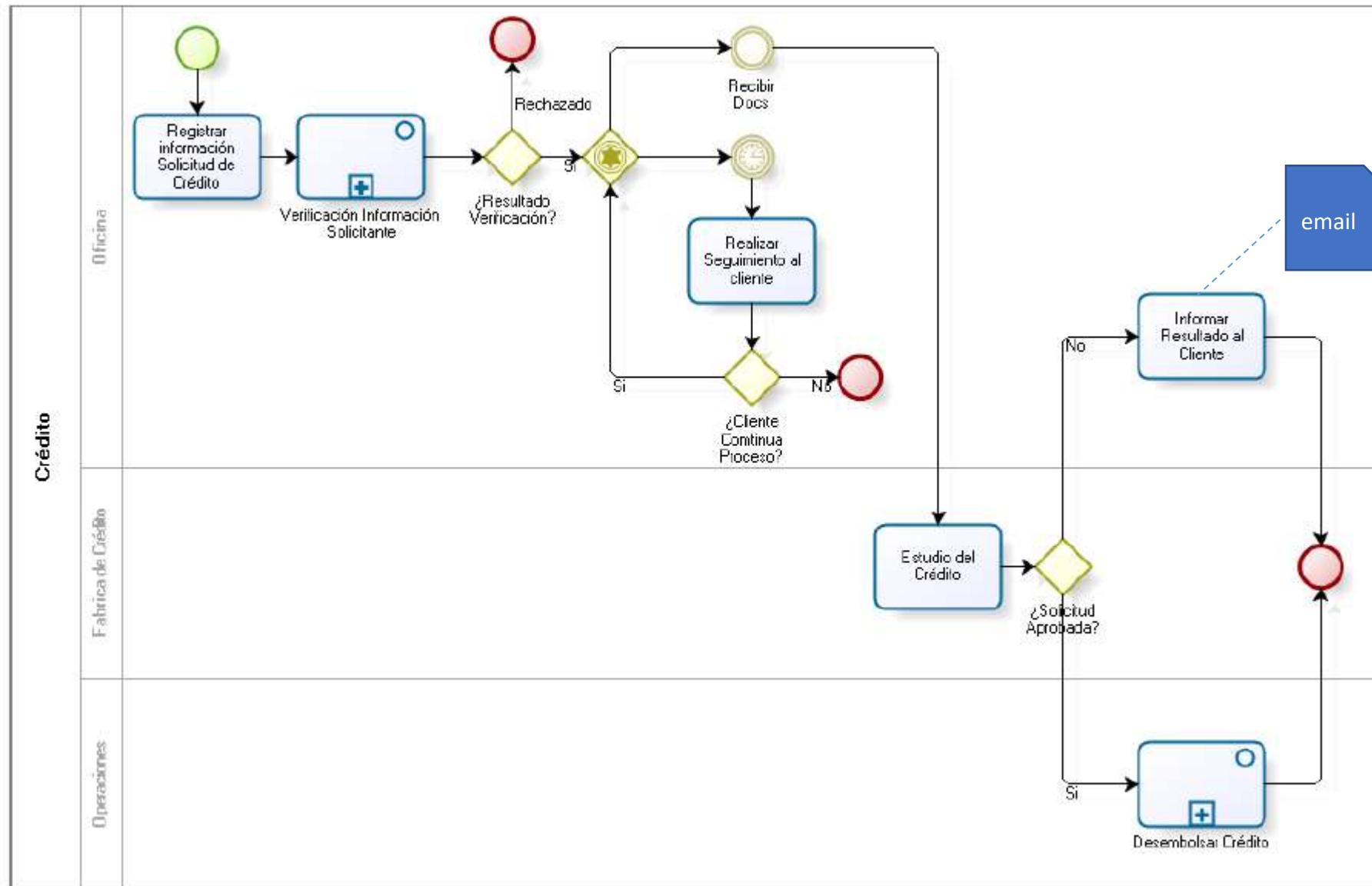


Comentario



Descripción

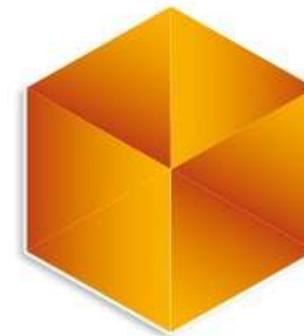
Desarrollo de Software I





Introducción a BPMN (Business Process Model and Notation)

Herramientas para modelar procesos de negocio en la notación BPMN





Introducción a BPMN (Business Process Model and Notation)

Videos complementarios

Resumen actividades, tares y subprocessos:

<https://www.youtube.com/watch?v=S3eEmkNzTxI>

Ejemplo de manejo de incidencias:

<https://www.youtube.com/watch?v=YOQf4anhTw>



Preguntas ?



Equipos para el proyecto del curso y exposiciones:

Registrar equipos del proyecto





Proyecto del curso



Desarrollo I

Economy of the
European Union

Gracias

