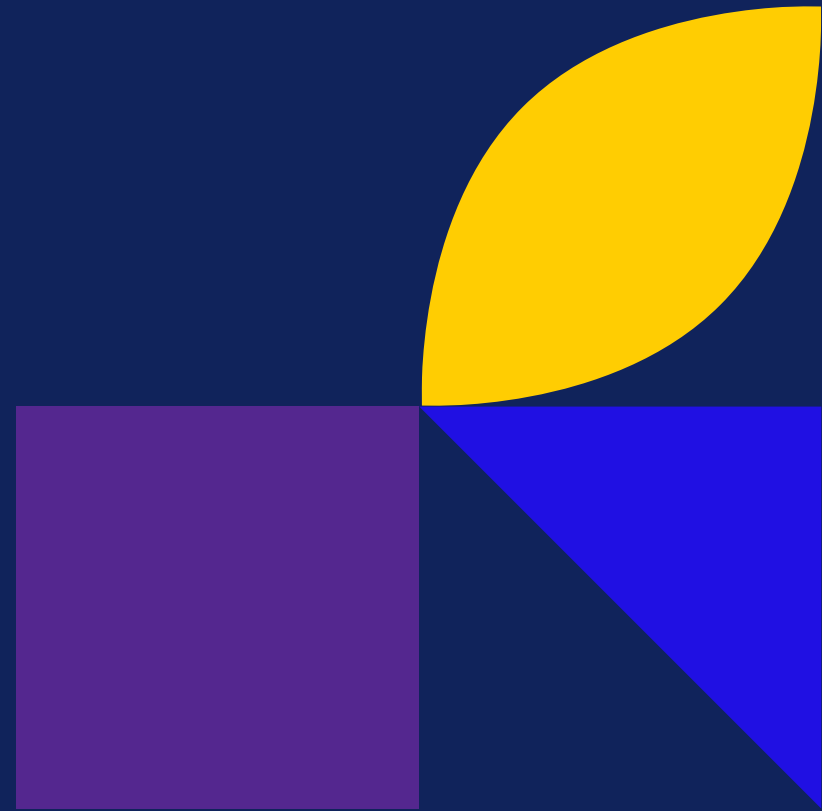


Módulo 2.

Aplicación de tecnologías emergentes en proyectos

Innovación tecnológica y tendencias emergentes en la ingeniería informática

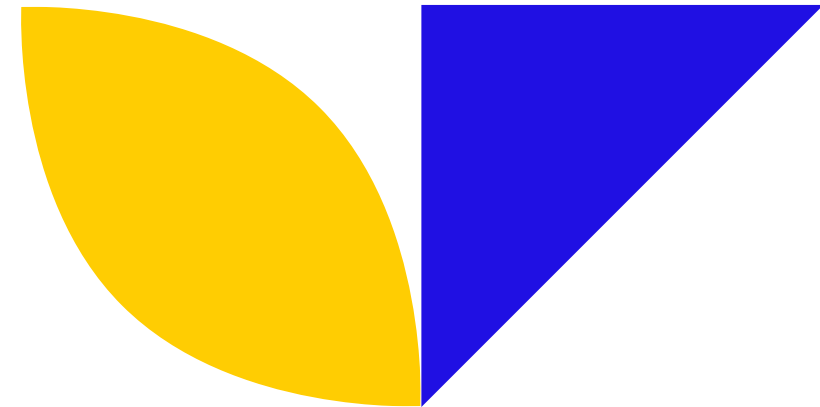


Índice

Integración de tecnologías emergentes en proyectos existentes

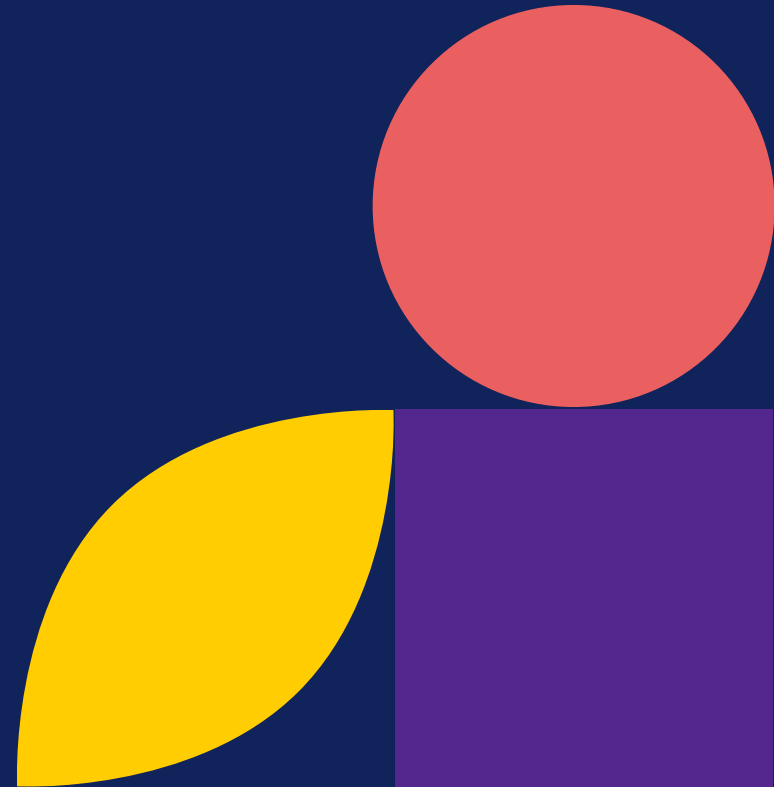
Herramientas y metodologías para la innovación

Casos de estudio y análisis de proyectos exitosos



01

Integración de tecnologías emergentes en proyectos existentes



01.1

Introducción



Introducción

La transformación digital: una nueva realidad

- En el módulo anterior hemos conocido las principales tendencias tecnológicas y las posibilidades de innovación e impacto que éstas pueden generar en nuestra sociedad.
- En este módulo vamos a centrarnos en qué debemos tener en cuenta a la hora de implantar una nueva tecnología en un negocio o proyecto ya existente.
- Conoceremos herramientas y metodologías que fomenten la innovación en nuestras organizaciones.
- Veremos también casos de estudio de proyectos exitosos, y por qué no, de proyectos que fracasaron.



Introducción

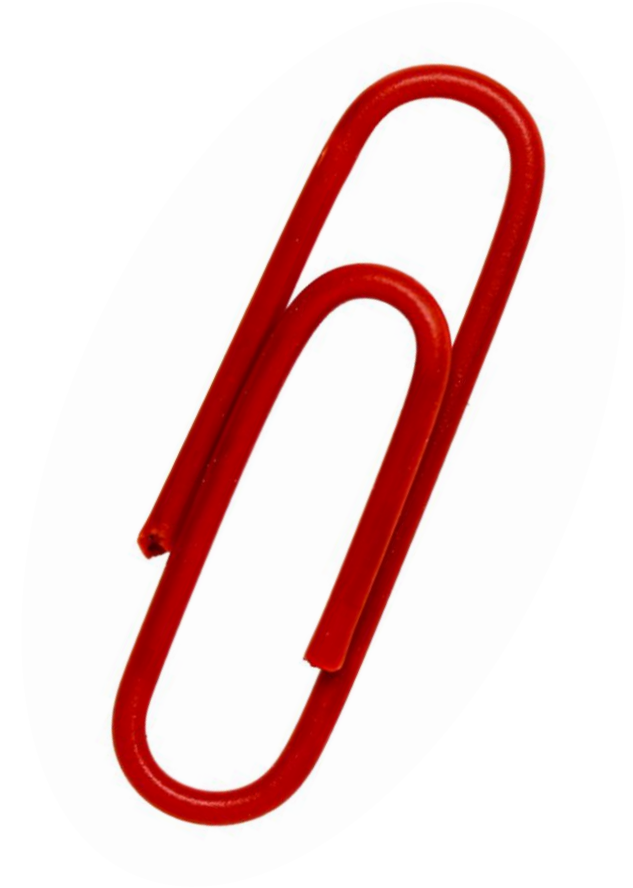
Protocolo reset: El test del clip

El consultor solitario

Silos

La red de conocimiento

**Conexión con la
innovación**



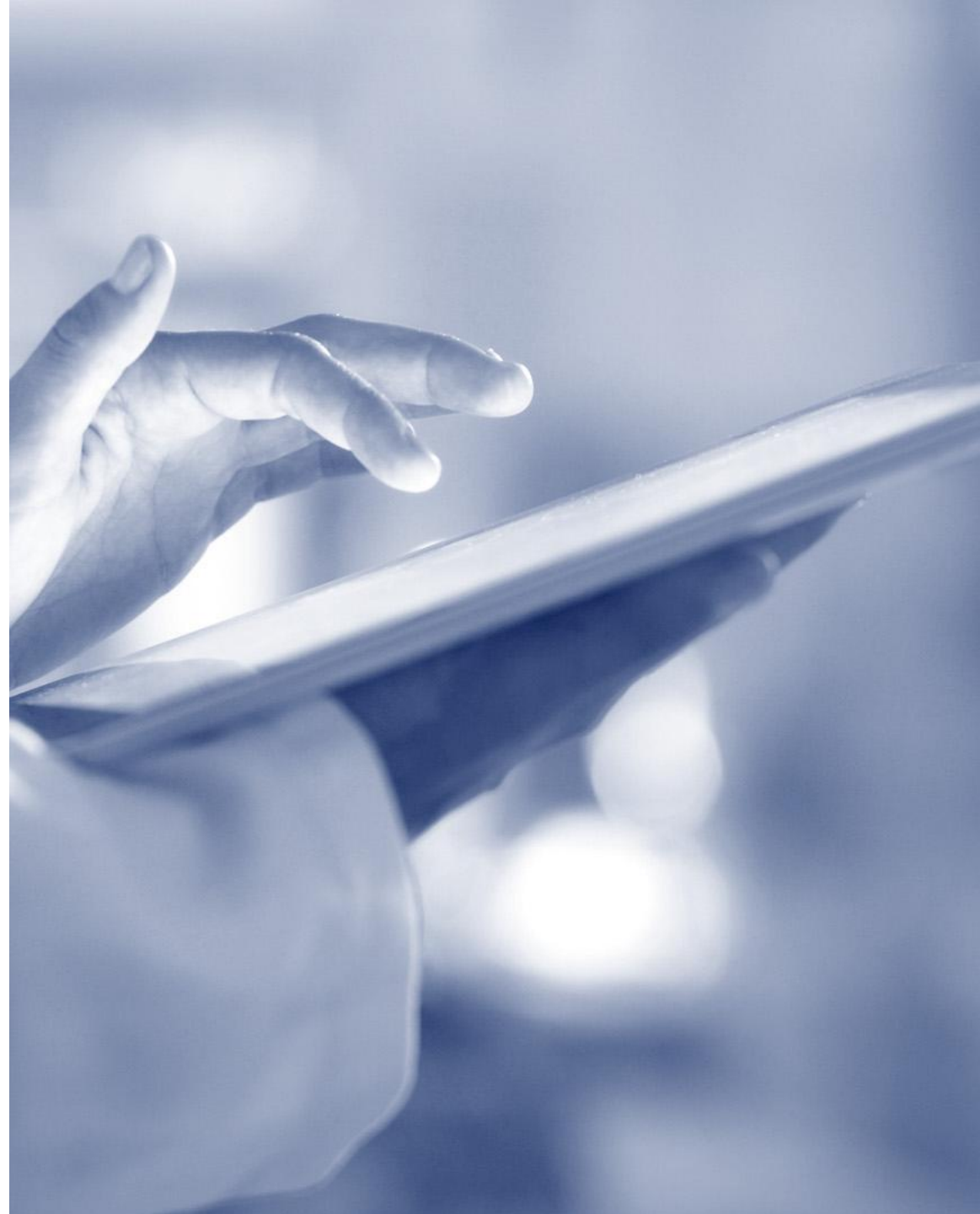
“Ninguno de nosotros es tan inteligente
como todos nosotros juntos”

Introducción

El mundo cambia rápido

- El mundo en constante cambio tecnológico.
- Gran aceleración.
- Nuevas oportunidades.

La aceleración tecnológica de los últimos años ha multiplicado las opciones disponibles, pero no ha ampliado la capacidad real de las empresas para absorberlas.

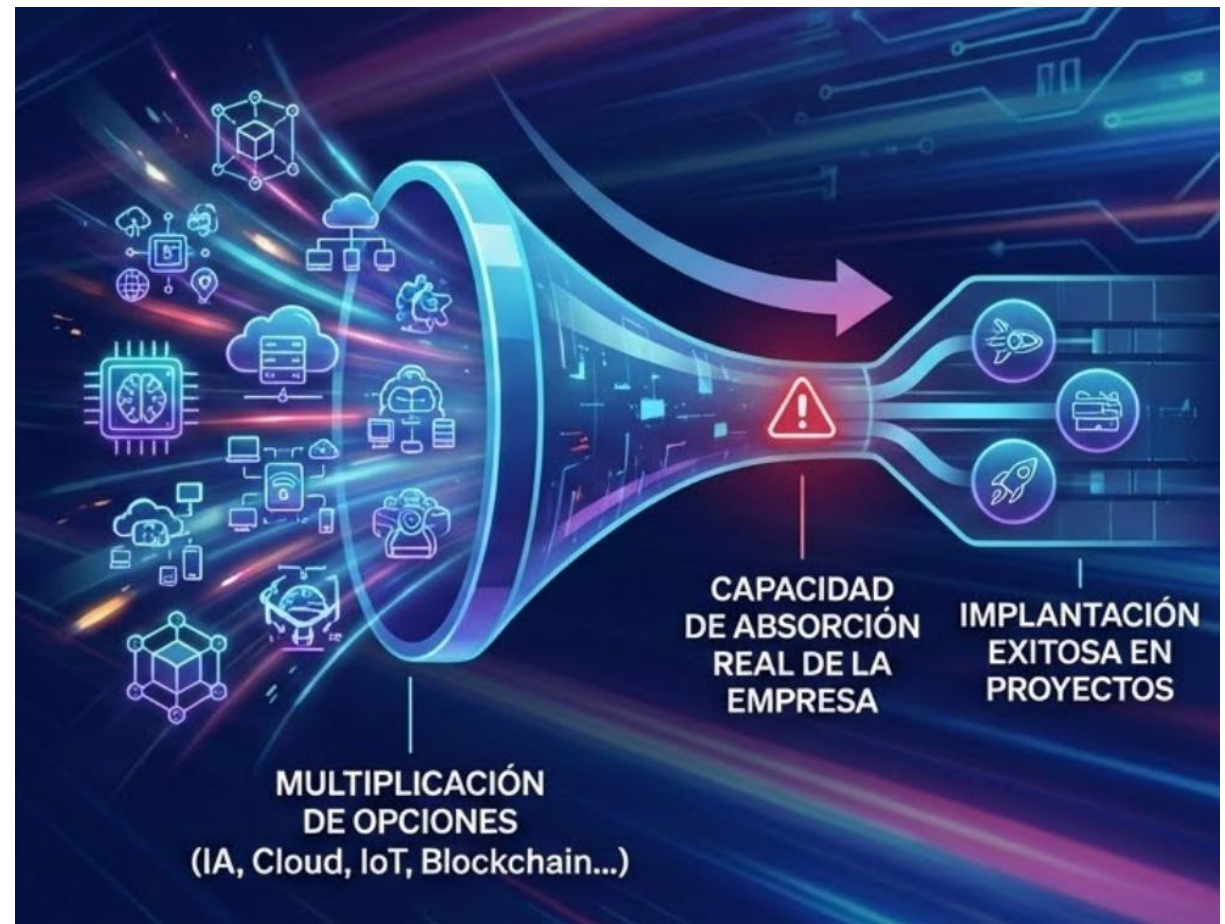


Introducción

No debemos caer en el hype tecnológico

- Exceso de expectativas.
- Hype tecnológico.
- Alineación con el negocio.

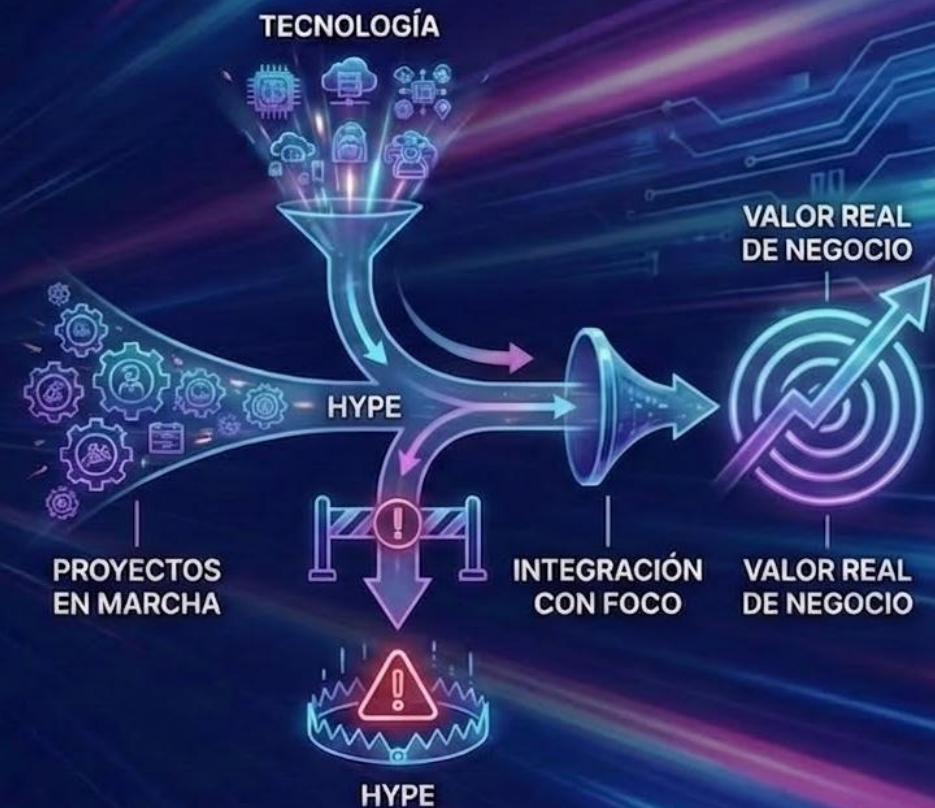
Uno de los grandes retos no es crear proyectos nuevos, sino **integrar tecnología en proyectos ya en marcha** sin caer en el "hype" y centrándonos en el valor real de negocio.



Introducción

Adopción tecnológica responsable

Queremos aprovechar las oportunidades, pero siendo realistas y valorando si la implantación en un negocio o proyecto concreto tiene posibilidades de éxito.



Introducción

Hoja de ruta de la unidad 1

01

Tendencias tecnológicas

Observatorio tecnológico
Queremos identificar las oportunidades que ofrece las nuevas tendencias tecnológicas a nuestro negocio.

02

Evaluar la viabilidad

Evaluar la viabilidad de éxito de implantar estas nuevas tendencias tecnológicas en el negocio.

03

Laboratorio de validación

Antes de una implantación en el propio negocio queremos hacer pruebas de validación.

04

Implantación y escalado

Decidimos implantar la nueva tecnología teniendo en cuenta los riesgos que podemos encontrar en el camino.

01.2

Identificación de oportunidades tecnológicas



Identificación de oportunidades tecnológicas

Detectar la necesidad

En esta profesión debemos estar al día, debemos estar atentos a las tecnologías emergentes que van apareciendo y poner la mirada en lo que estas pueden aportar en nuestro entorno, en nuestra organización, en nuestro negocio.

La objetividad metodológica como antídoto al entusiasmo subjetivo.

La tecnología no es un trofeo ni un fin en si mismo, sino una palanca de cambio que debe ser auditada con rigor.



Identificación de oportunidades tecnológicas

Dónde poner la mirada

- Estar atentos a las tendencias tecnológicas.
- Conocer las necesidades del negocio.
- Qué tendencias se están implantando en el sector.

Hay que tener el radar en marcha y estar atentos a lo que ocurre dentro y fuera.

Es importante conocer las tecnologías emergentes y valorar que pueden aportarnos.

Antes de introducir una tecnología hay que detectar necesidades dónde pueda aportar valor en el proyecto o negocio.

Es importante conocer el negocio, las necesidades de mejora, los cuellos de botella.

Identificación de oportunidades tecnológicas

Atentos a las tecnologías emergentes

Pero tenemos que aprender a evaluar estas oportunidades con objetividad. Antes siquiera de proponer cualquier integración, Podemos plantearnos una serie de preguntas filtro.

¿Es relevante?

¿Aporta un valor diferencial al proyecto existente o es solo una mejora cosmética?

¿Está madura?

¿Existe documentación, comunidad y soporte, o seremos los "beta testers" de un proveedor?

¿Tiene posibilidades de éxito?

Aquí entra la Capacidad de Absorción:
¿Mi equipo actual puede aprender esta tecnología? ¿Mi arquitectura legacy la soporta sin romperse?

Identificación de oportunidades tecnológicas

Technology Radar

El Radar Tecnológico de Thoughtworks es una publicación semestral que ofrece una visión sobre herramientas, técnicas, plataformas, lenguajes y frameworks. Esta herramienta de intercambio de conocimiento se basa en la experiencia de nuestros equipos globales y destaca elementos que podrías querer explorar en tus proyectos.



<https://www.thoughtworks.com/es-es/radar>

The Thoughtworks logo, consisting of a red slash followed by the word "thoughtworks" in lowercase.

Lo que hacemos AI/works™ Con quién trabajamos **Insights** Ingeniería Carreras Acerca Contacto

Radar Tecnológico Vol 33

Volumen 33 | Noviembre 2025

Radar Tecnológico

Una guía de opinión sobre el entorno tecnológico actual

Susíbete

Identificación de oportunidades tecnológicas

Cómo leer el radar

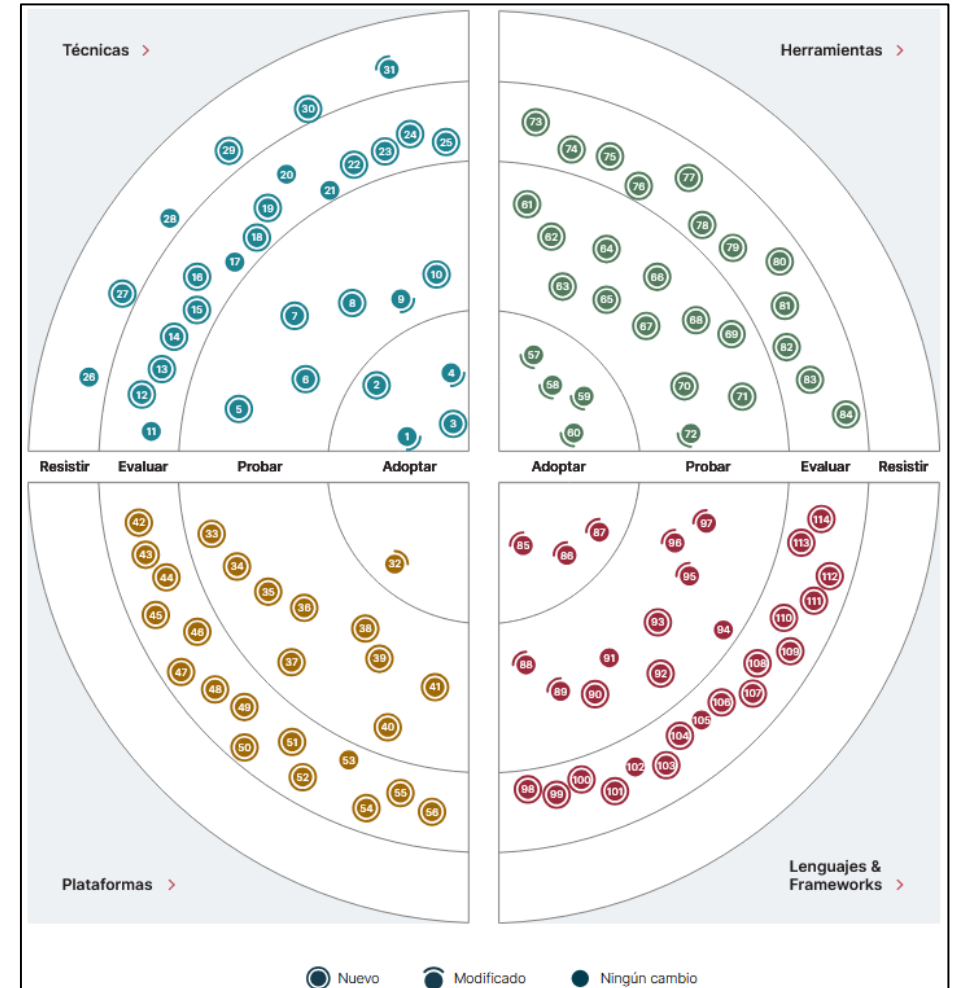
Cada insight que compartimos está representada por un blip. Algunos blips son nuevos en este volumen del Radar, otros cambian de anillo cuando nuestra recomendación se actualiza.

Los anillos son:

- Adoptar: Blips que creemos deberías considerar seriamente utilizar.
- Probar: Elementos que creemos que ya están listos para usarse, pero que aún no están tan consolidados como los del anillo Adoptar.
- Evaluar: Elementos que vale la pena observar de cerca, pero que quizá aún no sea momento de probar, a menos que encajen particularmente bien contigo.
- Resistir: Avanza con precaución.



<https://www.thoughtworks.com/es-es/radar>



Identificación de oportunidades tecnológicas

Actividad: enciende el radar

Dinámica: Exploración de tecnologías disponibles.

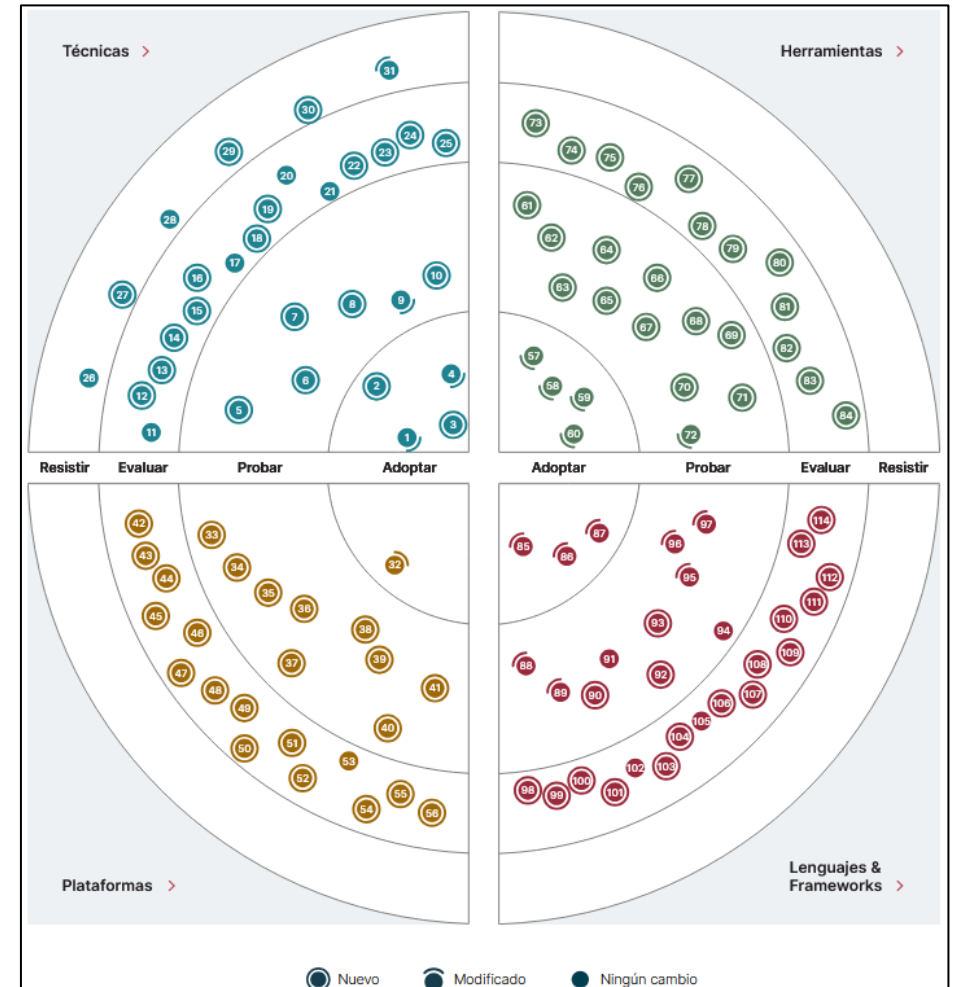
Objetivo: Familiarizarse con el Rada.

Acción: Que los alumnos aprendan a navegar por el Radar de Thoughtworks para identificar oportunidades tecnológicas, diferenciando entre el "ruido" mediático y las herramientas que realmente pueden resolver problemas de negocio en un entorno realista.



<https://www.thoughtworks.com/es-es/radar/search>

<https://www.thoughtworks.com/es-es/radar>



Identificación de oportunidades tecnológicas

Actividad: construye tu propio radar

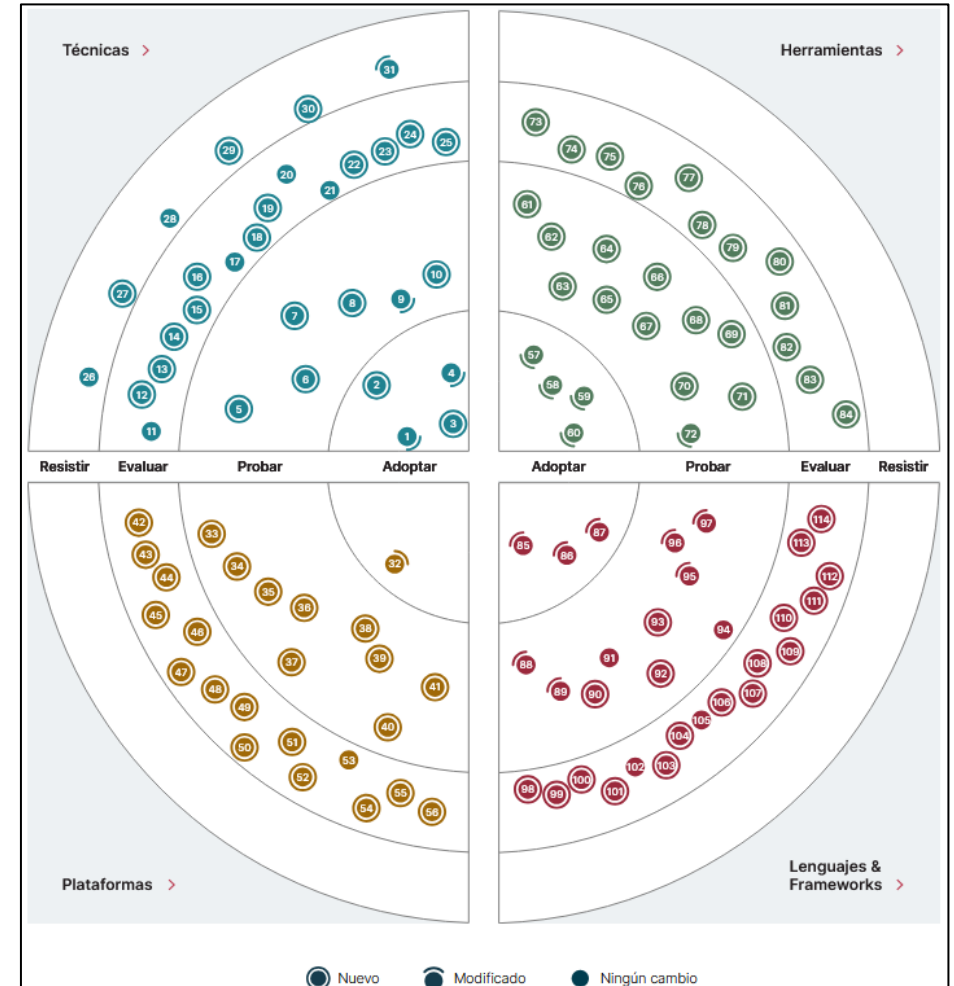
Dinámica: Exploración de tecnologías disponibles.

Objetivo: "Poner blanco sobre negro" el estado tecnológico de la entidad, priorizando las inversiones en innovación y detectando riesgos de obsolescencia técnica mediante un consenso colaborativo sobre el ciclo de vida de las tecnologías.



<https://www.thoughtworks.com/es-es/radar/byor>

<https://www.thoughtworks.com/es-es/radar>



01.3

Evaluar la viabilidad



Evaluar la viabilidad

Antes de implementar la tecnología

Cuando estamos hablando de la implantación de tecnologías emergentes en negocios o proyectos existentes donde tenemos un presupuesto acotado, donde el retorno se exige a corto plazo, la cuestión no es integrar una nueva tecnología, sino si va a aportar un valor al negocio, y en qué momento tiene sentido invertir en ellas.

El coste de oportunidad de una tecnología emergente no es solo su implementación, sino el riesgo de inestabilidad en un sistema que ya factura.



“La tecnología es un medio, no un fin”

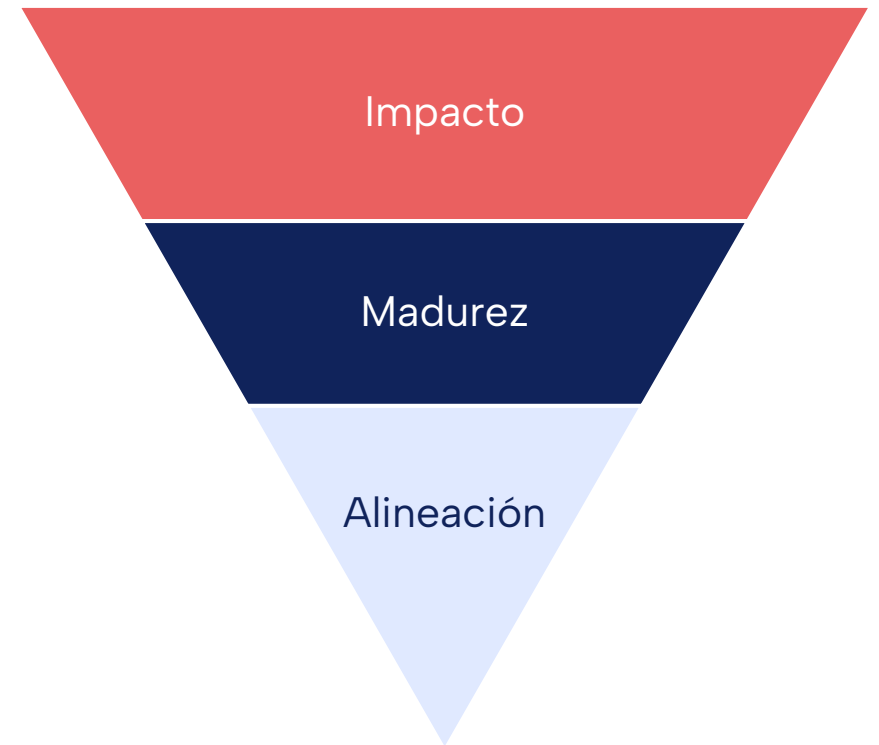
Evaluar la viabilidad

El arte de decir No (o "Todavía no") a la tecnología de moda.

Al final tenemos que tomar decisiones, pero estas que van más allá de un listado de tendencias o modas tecnológicas en el mercado.

Hay que valorar otros parámetros siguiendo una metodología que nos permitan decidir de forma objetiva y analizar cómo introducir una tecnología emergente en un proyecto real sin que el proyecto fracase.

A continuación, descubriremos algunos métodos que nos ayudarán a valorar las tecnologías emergentes a la hora de decidir implantarla en un proyecto real.



Evaluar la viabilidad

Cuadro de decisión I-M-A

El ingeniero pragmático no sigue tendencias; calcula el riesgo de la novedad frente a la certeza del valor.

Este sencillo marco de trabajo, objetiva la decisión y protege la estabilidad del proyecto.

Según el Impacto, la Madurez y la Alineación con los objetivos de la entidad, podemos decidir: Adoptar, preparar o quedarnos en observación.

Decisión	Escenario típico (I-M-A)	Acción para el ingeniero/lead técnico
Adoptar	Alto impacto Estable madurez Total alineamiento	Iniciar el plan de integración en producción inmediatamente. Es una ventaja competitiva clara y segura.
Preparar	Alto/medio impacto Adolescente madurez Parcial/total alineamiento	Iniciar una poc (prueba de concepto) o un piloto en un módulo no crítico. Formar al equipo. No va a producción mañana, pero debemos estar listos.
Observar	Bajo impacto (hoy) Cualquier madurez Cualquier alineamiento ...O madurez experimental con presupuesto acotado	Monitorear la tecnología. Leer whitepapers, seguir la comunidad. No invertir recursos económicos ni tiempo de desarrollo aún. Reevaluar en el próximo ciclo.

01.4

Impacto



Impacto

Impacto estratégico → valor para la organización

- ¿Vale la pena esta tecnología para la organización? → impacto estratégico.
- El **impacto estratégico** evalúa **qué valor puede generar una tecnología para la organización o el proyecto**.
- No todas las tecnologías emergentes son relevantes para todas las organizaciones. Una tecnología puede ser muy avanzada pero **no aportar valor estratégico en un contexto concreto**.
- Si una tecnología emergente vemos que no va a aportar valor a la organización ya no se debe valorar.
- Para **detectar las tecnologías que aportan valor** es importante la fase de observación de estas tecnologías, que valor están aportando en otros entornos y valorar si pueden aportar valor en nuestros proyectos.



Impacto

Qué mejoras nos aporta una tecnología emergente

¿Qué se analiza?

Normalmente se evalúan factores como:

Impacto en el modelo de negocio

- Nuevos mercados
- Nuevos productos
- Nuevos modelos de servicio

Impacto en el cliente

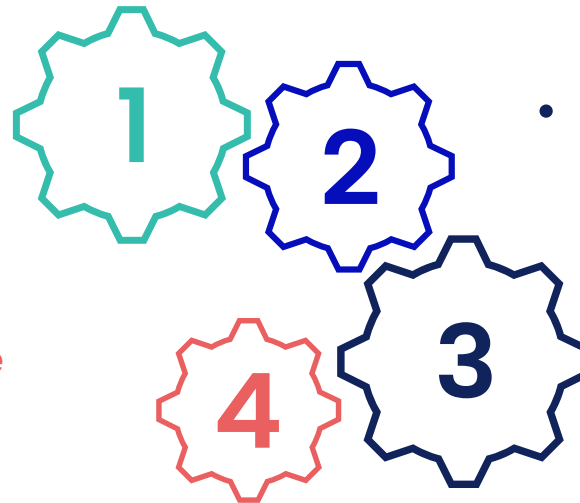
- Mejor experiencia de usuario
- Nuevos canales o servicios

Ventaja competitiva

- ¿Permite diferenciarse de los competidores?
- ¿Permite ofrecer nuevos productos o servicios?

Impacto en el negocio

- Aumento de ingresos
- Reducción de costes
- Mejora de eficiencia



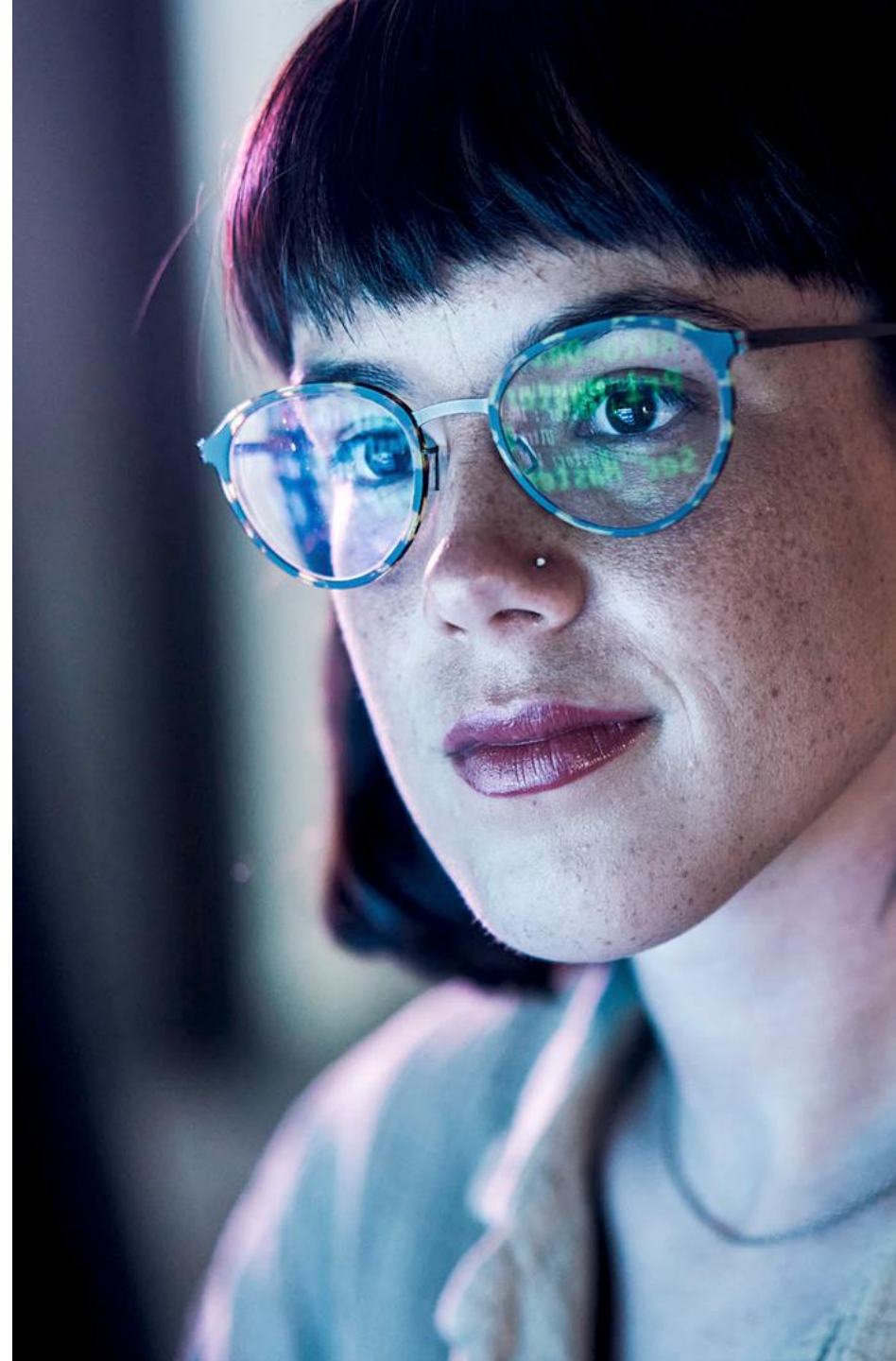
Impacto

Competitive Advantage (Michael Porter)

1. Rivalidad entre competidores existentes.
2. Amenaza de nuevos entrantes.
3. Poder de negociación de los proveedores.
4. Poder de negociación de los clientes.
5. Amenaza de productos sustitutos.



El caso de **Netflix**. No ganaron por usar la nube (tecnología), sino porque usar la nube les permitió escalar su catálogo globalmente de una forma que los videoclubs (sustitutos) no podían igualar. Transformaron el poder del cliente (fuerza 4) dándole control total sobre el catálogo.



Impacto

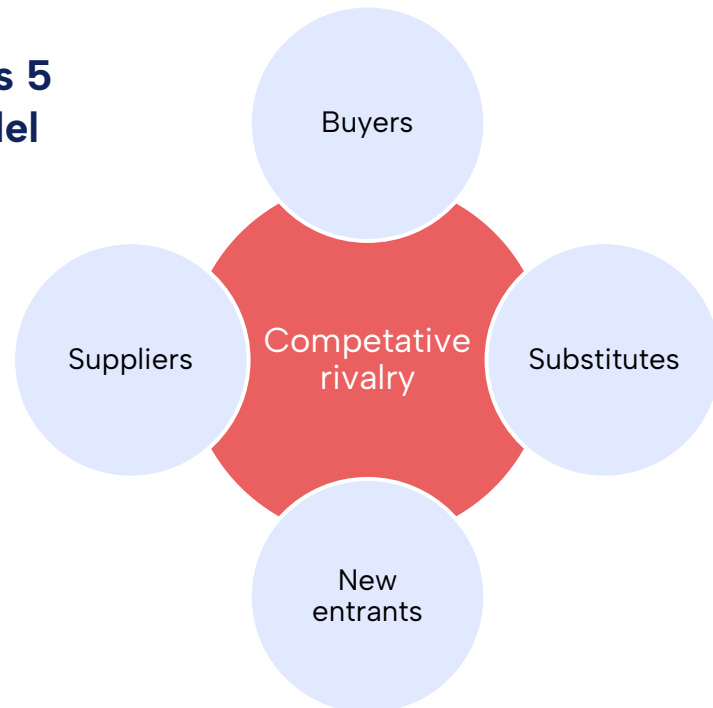
Ejemplo. Competitive Advantage (Michael Porter)

Fuerza de Porter	Criterio de Evaluación	Nivel de Impacto	Descripción del Impacto en el Negocio
1. Rivalidad competitiva	¿La tecnología nos diferencia de la competencia actual?	Medio	Mejora procesos internos (eficiencia), pero los competidores pueden copiarlo rápido.
2. Nuevos entrantes	¿Crea una barrera de entrada técnica difícil de replicar?	Alto	Genera una ventaja competitiva basada en IP propia o datos difíciles de obtener.
3. Poder de proveedores	¿Nos libera de la dependencia de un solo proveedor (Lock-in)?	Bajo	El impacto es limitado si solo cambiamos una licencia por otra suscripción SaaS.
4. Poder de clientes	¿Aumenta el coste de cambio para nuestros clientes?	Muy Alto	Transformación del negocio: El cliente percibe tanto valor que no puede irse a la competencia.
5. Productos sustitutos	¿Protege nuestro mercado de soluciones alternativas?	Alto	La tecnología hace que nuestra solución sea mucho más barata o rápida que los sustitutos.

Actividad. El tablero estratégico de Porter (opcional)

Objetivo: analizar si la tecnología elegida nos da fuerza frente al mercado o si nos genera riesgos de dependencia para el caso Residencia Horizonte.

The Porter's 5 Forces Model



01.5

Madurez



Madurez

Evaluar la madurez de tecnologías emergentes

Métodos y herramientas para evaluar la madurez:

- Technology Radar.
- TRL (Technology Readiness Level).
- Modelo Gartner Hype Cycle.



Al final tenemos que tomar decisiones, pero estas que van más allá de un listado de tendencias o modas tecnológicas en el mercado.



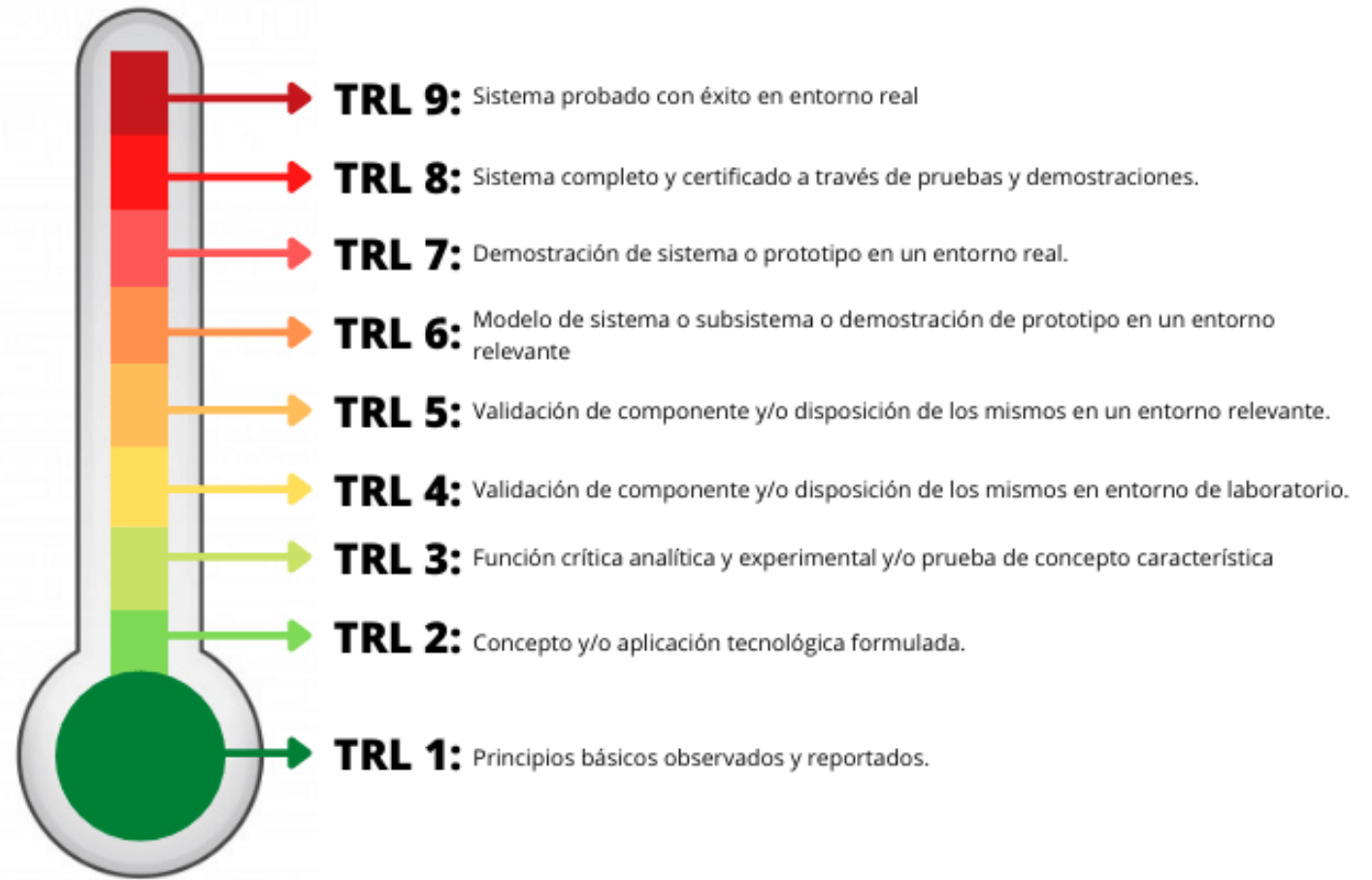
Madurez

TRL (Technology Readiness Level)

TRL (**T**echnology **R**eadiness **L**evel) es una herramienta muy usada en innovación desarrollado por la NASA.

El TRL es una métrica de gestión de la innovación, que permite evaluar de forma objetiva qué tan cerca está una tecnología de ser desplegada en un entorno real.

La decisión de implementar depende del nivel de TRL en relación con el riesgo del proyecto. TRL Tiene 9 niveles que veremos continuación.



Madurez

Los 9 niveles del TRL

Fase de investigación

Niveles 1-3

- **TRL 1:** Observación de principios básicos.
- **TRL 2:** Formulación del concepto tecnológico.
- **TRL 3:** Prueba de concepto experimental (PoC inicial).

Fase de desarrollo

Niveles 4-6

- **TRL 4:** Validación tecnológica en entorno de laboratorio.
- **TRL 5:** Validación tecnológica en un entorno relevante (simulado).
- **TRL 6:** Demostración del modelo o prototipo en un entorno relevante.

Fase de despliegue e implantación

Niveles 7-9

- **TRL 7:** Demostración de prototipo en entorno operativo real.
- **TRL 8:** Sistema completo y certificado a través de pruebas y demostraciones.
- **TRL 9:** Sistema real probado con éxito en un entorno operativo (misión cumplida).

Madurez

Cómo valoro el nivel de TRL

Fase de investigación (TRL 1-3)

Se valora mediante la existencia de Whitepapers, publicaciones científicas revisadas por pares o pruebas de concepto (PoC).

Fase de desarrollo (TRL 4-6)

Se mide por la existencia de prototipos funcionales.

Fase de despliegue (TRL 7-9)

Aquí la valoración es binaria: ¿Está operando con usuarios reales?

Un TRL 7 implica un piloto con fuego real pero a pequeña escala, mientras que el TRL 9 se confirma cuando el sistema es el estándar operativo y ha demostrado estabilidad bajo estrés durante un tiempo prolongado.

Proyectos críticos (Core Business)

No deberíamos integrar nada que esté por debajo de un TRL 8 o 9. El riesgo de inestabilidad en un sistema que ya factura es demasiado alto.

Proyectos de innovación o módulos no críticos

Podemos experimentar con tecnologías en TRL 6 o 7, sabiendo que estamos en una fase de pilotaje.

Madurez

Modelo Gartner Hype Cycle

Cuando las nuevas **tecnologías hacen promesas audaces**, ¿cómo podemos distinguir lo que es una exageración de lo que es comercialmente viable? ¿Y cuándo se llegarán a amortizar estas soluciones, suponiendo que lo lleguen a hacer?

El Hype Cycle de Gartner es una representación gráfica en forma de curva que representa la madurez y adopción de las tecnologías y apps y cómo son potencialmente relevantes para resolver problemas comerciales reales y aprovechar nuevas oportunidades.

La metodología del Hype Cycle (literalmente, "ciclo de sobreexpectación" en español y a menudo llamado "curva de Gartner") de Gartner te permite ver cómo una tecnología o app evolucionará con el tiempo, y te proporciona una fuente sólida de información para que puedas gestionar su implementación dentro del contexto de tus objetivos de negocio específicos.



<https://www.gartner.es/es/metodologias/hype-cycle>



Madurez

El ciclo de expectativas (Hype Cycle)

Hype Cycle → expectativas de mercado.

Creado por la consultora Gartner.

Este modelo analiza **el ciclo de expectativas de una tecnología**.



“El mejor momento para entrar suele ser la "Rampa de Consolidación", no el "Pico de expectativas"



Madurez

Como nos ayuda a decidir la implantación

- **Utilidad:** análisis de tendencias tecnológicas
planificación estratégica
roadmaps de innovación.
- **Ventaja:** permite evitar invertir cuando la tecnología está en fase de "hype".



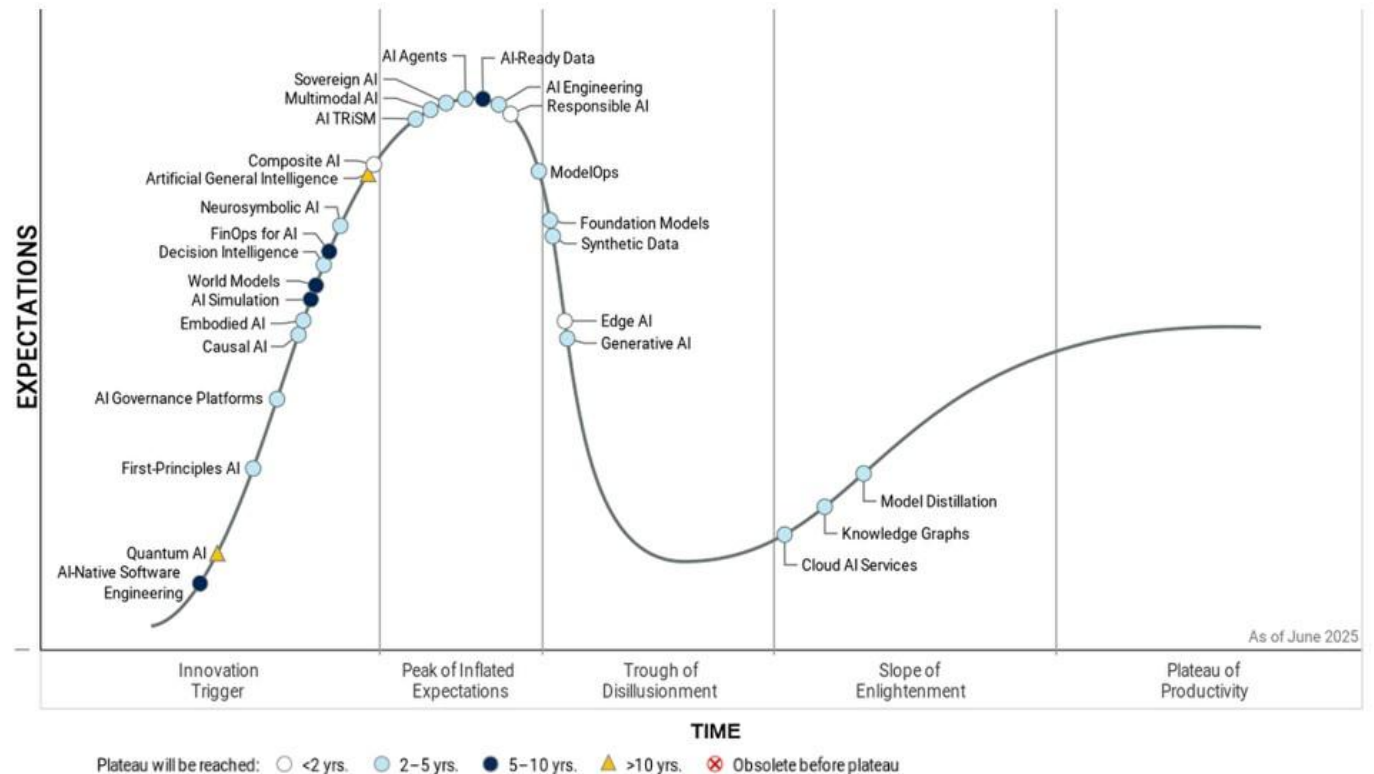
Madurez

¿En qué fase de Gartner se encuentra una tecnología?

La fuente principal y más fiable es la propia consultora **Gartner**, que publica anualmente más de 100 "Hype Cycles" específicos por industria y dominio tecnológico.



<https://www.gartner.es/es/articulos/hype-cycle-para-las-tecnologias-emergentes>



01.6

Alineamiento



Alineamiento

Alineamiento – capacidad de adopción

¿Podemos adoptar realmente esta tecnología?

- El **alineamiento organizativo** analiza **si la organización tiene la capacidad real para adoptar esa tecnología**.
- Una tecnología puede tener alto impacto, pero si la empresa **no tiene capacidades para implementarla**, el proyecto fracasará.



Alineamiento

Aspectos que debemos analizar

Normalmente se evalúan cuatro aspectos principales.



Capacidades tecnológicas

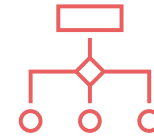
- Infraestructura tecnológica.
- Sistemas existentes.
- Integración con otras plataformas.

Ejemplo: Una empresa sin infraestructura cloud tendrá dificultades para aplicar IA.



Competencias del equipo

- Talento digital.
- Especialistas tecnológicos.
- Formación del personal.



Cultura organizativa

- Apertura a la innovación.
- Tolerancia al riesgo.
- Apoyo del liderazgo.

Muchas transformaciones digitales fracasan por resistencia al cambio.

Alineamiento

Aspectos que debemos analizar

4. Recursos disponibles

- Presupuesto.
- Tiempo.
- Apoyo directivo.



Ejemplo:

Tecnología: blockchain.

Empresa pequeña:

- sin expertos
- sin infraestructura
- sin casos de uso claros
- sin apoyo de la dirección

Resultado → abocada al fracaso

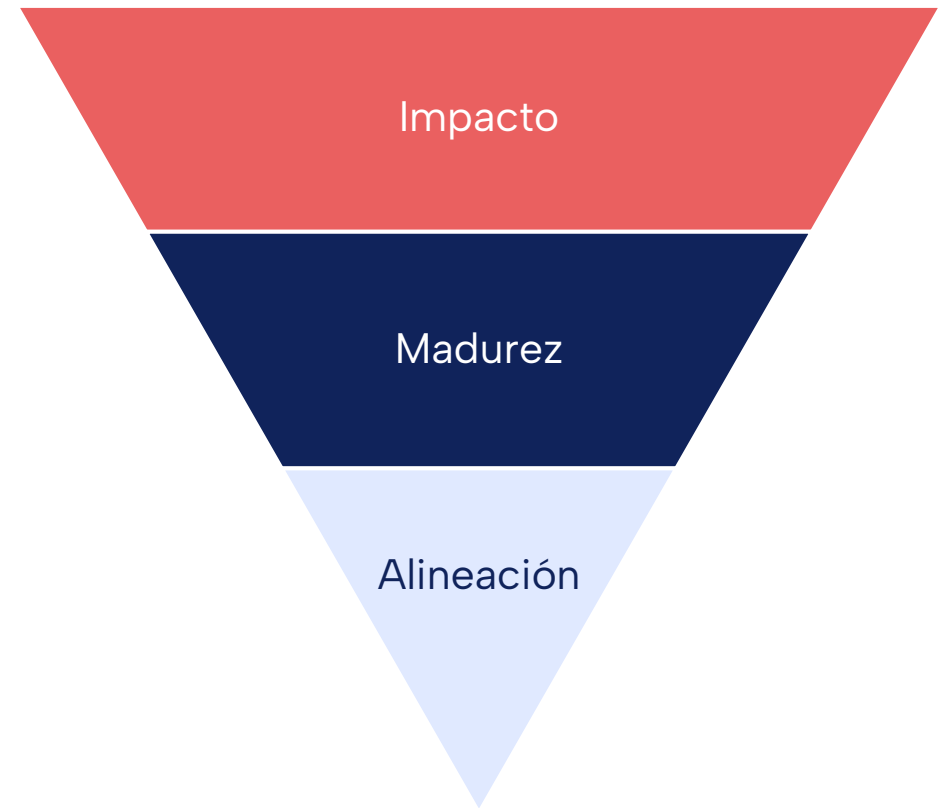


Alineamiento

Actividad: El Comité de Inversión Tecnológica

Objetivo: Tomar una decisión ejecutiva "en frío" sobre la implementación basándose en tres filtros técnicos:

Impacto, **M**adurez y **A**lineamiento.



01.7

Laboratorio de validación



Laboratorio de validación

Nuestro seguro de vida profesional

El objetivo aquí no es que "funcione", sino que "aporte valor en un entorno controlado" y despejemos dudas de implantación antes de decidir una implantación en un entorno real.



"Experimentación en entornos controlados: Del PoC al Piloto"



Laboratorio de validación

Equivócate rápido y barato

Fallo rápido (Fail Fast)

Un resultado negativo en un piloto es un éxito de gestión, porque evita una inversión ruinosa a mayor escala.



“Falla rápido y barato”

Las pruebas nos van a permitir:

- Validar la tecnología
- Comprobar impacto
- Reducir riesgo



Laboratorio de validación

Decidir el ámbito de aplicación

La entrada de una nueva tecnología puede prepararse para aplicarse en:

Un proceso
específico.

Un departamento
concreto.

Una fase
determinada del
proyecto
existente



Laboratorio de validación

Prepara el laboratorio

El concepto de **laboratorio** implica un **método científico**: hipótesis, entorno controlado, medición y conclusiones.

Sandbox Tecnológico

Creamos un **entorno seguro** para reducir la incertidumbre técnica y de negocio.



Laboratorio de validación

Tipos de instrumentos en el laboratorio

Debemos diferenciar claramente los instrumentos que utilizaremos en nuestro "Sandbox", para poder elegir la herramienta que mejor responda a la incertidumbre que queremos despejar, optimizando tiempo y presupuesto.

PoC
(Prueba de
Concepto)

Prototipo

MVP
(Producto
Mínimo
Viable)

Proyecto
Piloto

Laboratorio de validación

PoC (Prueba de concepto)

Es el instrumento más básico y se centra exclusivamente en la **factibilidad técnica**:



Objetivo: responder a la pregunta: "¿Es técnicamente posible?".



Uso real: validar integraciones críticas, como conectar un sensor moderno a una base de datos antigua (*legacy*).



Alcance: no suele tener interfaz de usuario ni código optimizado; es código descartable para validar una hipótesis técnica.



Laboratorio de validación

Prototipo

El prototipo nos permite validar de la experiencia y forma.



Objetivo

Validar la interacción. "¿Cómo interactuará el operario o usuario con esta interfaz?".



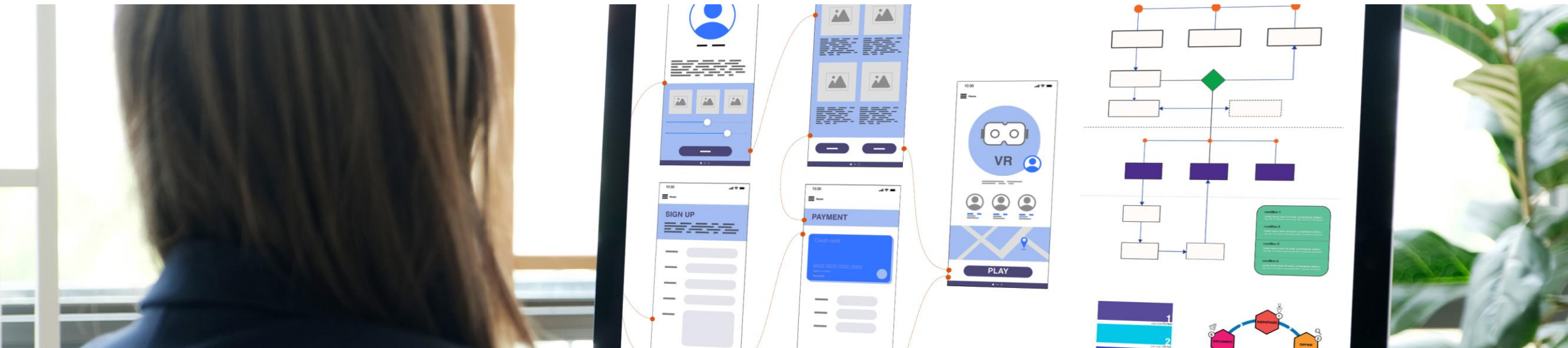
Uso real

Modelar el flujo de trabajo de una nueva tecnología de realidad aumentada antes de programar la lógica completa.



Alcance

Simula el comportamiento del sistema final para recoger *feedback* temprano sobre usabilidad.



Laboratorio de validación

MVP (Producto Mínimo Viable)

Aquí entramos en la validación del valor de negocio.



Objetivo

Descubrir si la funcionalidad mínima aporta valor real.
"¿Pagaría el cliente o ahorraría la empresa con esta funcionalidad mínima?"

Uso real

Lanzar una versión simplificada de una herramienta de análisis de datos para un solo departamento.

Alcance

Es un producto funcional que contiene el conjunto mínimo de características para ser utilizable y medible.

Laboratorio de validación

Proyecto piloto

Es la prueba de fuego: la validación en un entorno real controlado.



Objetivo

Medir el impacto operativo en condiciones de producción.



Uso real

Se acota la aplicación de la solución a un grupo reducido de usuarios reales dentro de la organización antes del despliegue masivo.



Alcance

Implementación limitada pero formal que permite observar cómo la tecnología altera los procesos y el día a día.

Laboratorio de validación

Guía de selección del método de validación

Cuadro de selección de método de validación en laboratorio:

Instrumento	Objetivo	¿Qué validamos?
PoC	Factibilidad técnica	"¿Se puede conectar?"
Prototipo	Experiencia y forma	"¿Cómo se usa?"
MVP	Valor de negocio	"¿Sirve para ganar/ahorrar dinero?"
Piloto	Impacto operativo	"¿Sobrevive al uso real diario?"

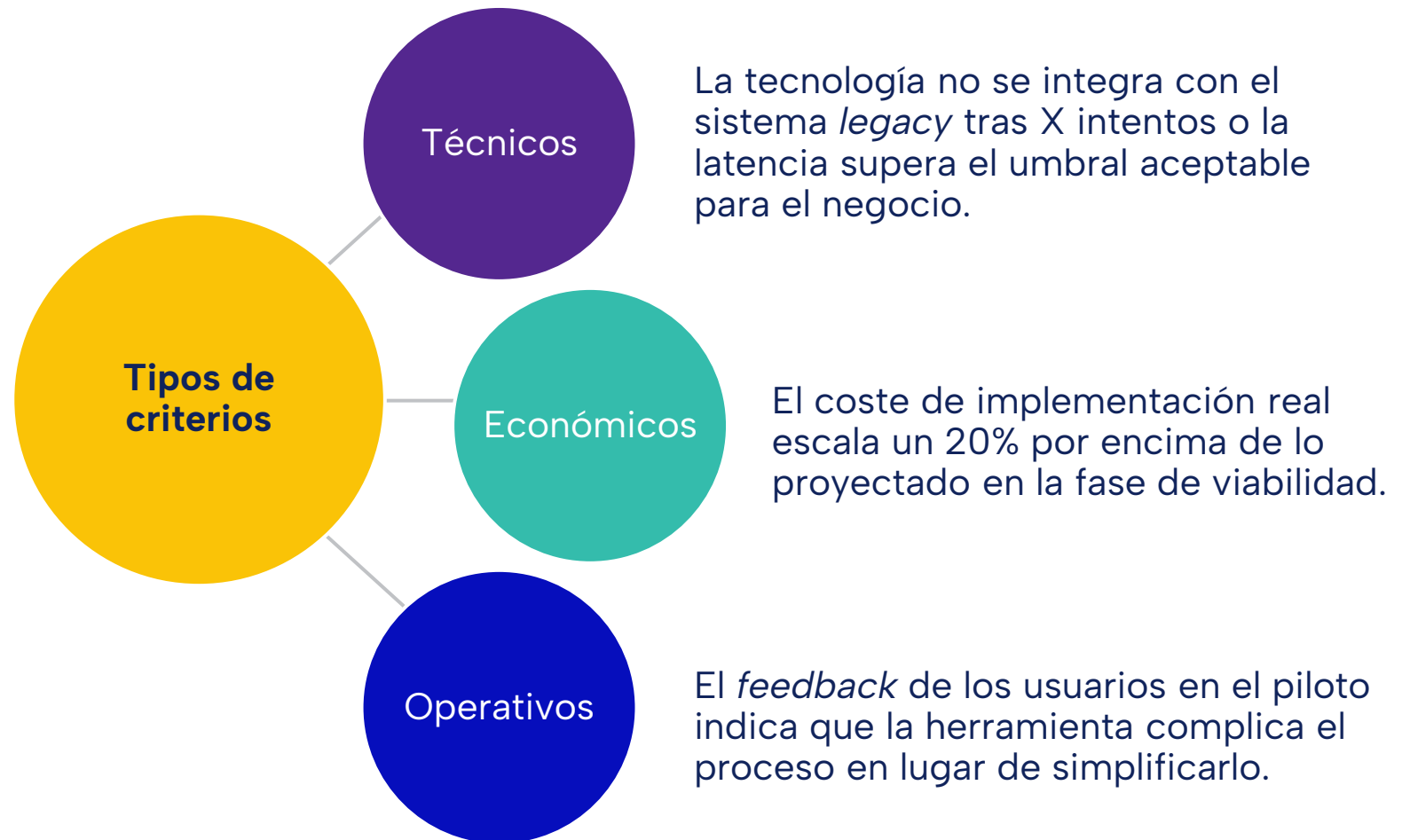


Laboratorio de validación

Criterios de parada

Los criterios de parada son métricas o condiciones predefinidas que, de no cumplirse, obligan a detener el experimento de inmediato.

Estos deben fijarse **antes** de empezar cualquier PoC o Piloto para mantener la imparcialidad.



Laboratorio de validación

Falacia del costo hundido

Establecer **criterios de parada (Stop/Go Decisions)** es fundamental en un "Laboratorio de Validación" para evitar lo que en gestión de proyectos llamamos la "**falacia del costo hundido**": seguir invirtiendo en algo que no funciona solo porque ya hemos gastado tiempo y dinero.

Queremos evitar la "patada adelante"



Laboratorio de validación

El concepto de "Pivotar" o golpe de timón

"Para un proyecto no es un fracaso, es una victoria de la gestión".

Si un criterio de parada se activa, tienen dos opciones:

A

Kill

Cerrar el proyecto y documentar las lecciones aprendidas (Post-mortem).

B

Pivot

Cambiar el enfoque. "La tecnología A no sirve para este problema, pero hemos descubierto que serviría para este otro".

Actividad. El juicio del proyecto (opcional)

Objetivo: aprender a identificar cuándo un proyecto de tecnología emergente debe detenerse (Kill) o cambiar de rumbo (Pivot) antes de que sea un fracaso costoso.



01.8

Implantación y escalado



Implantación y escalado

El paso a producción

Una vez que el "Laboratorio" da luz verde, se presentan los resultados y se decide la integración en el negocio de la nueva tecnología.

En esta fase el reto es el **paso a producción**.

Una disrupción sin planificación genera caos en la organización.

En esta fase, el éxito no depende solo del código, sino de cómo la tecnología se asienta en la estructura viva de la empresa.

La implantación y el escalado requieren un enfoque multifacético en el que se debe tener en cuenta:

1. **Planificación estratégica.**
2. **Arquitectura de integración.**
3. **Gestión del cambio organizacional.**
4. **Monitorización y gobernanza.**



Implantación y escalado

Disrupción sin planificación, caos en la organización

Riesgos de la implantación de cualquier proyecto, acentuado por nuevos factores.

Los cambios que pueden generar las nuevas tendencias tecnológicas pueden alterar procesos, herramientas y la forma de trabajar de las personas.

Hay que gestionar este cambio en la implantación de forma adecuada.



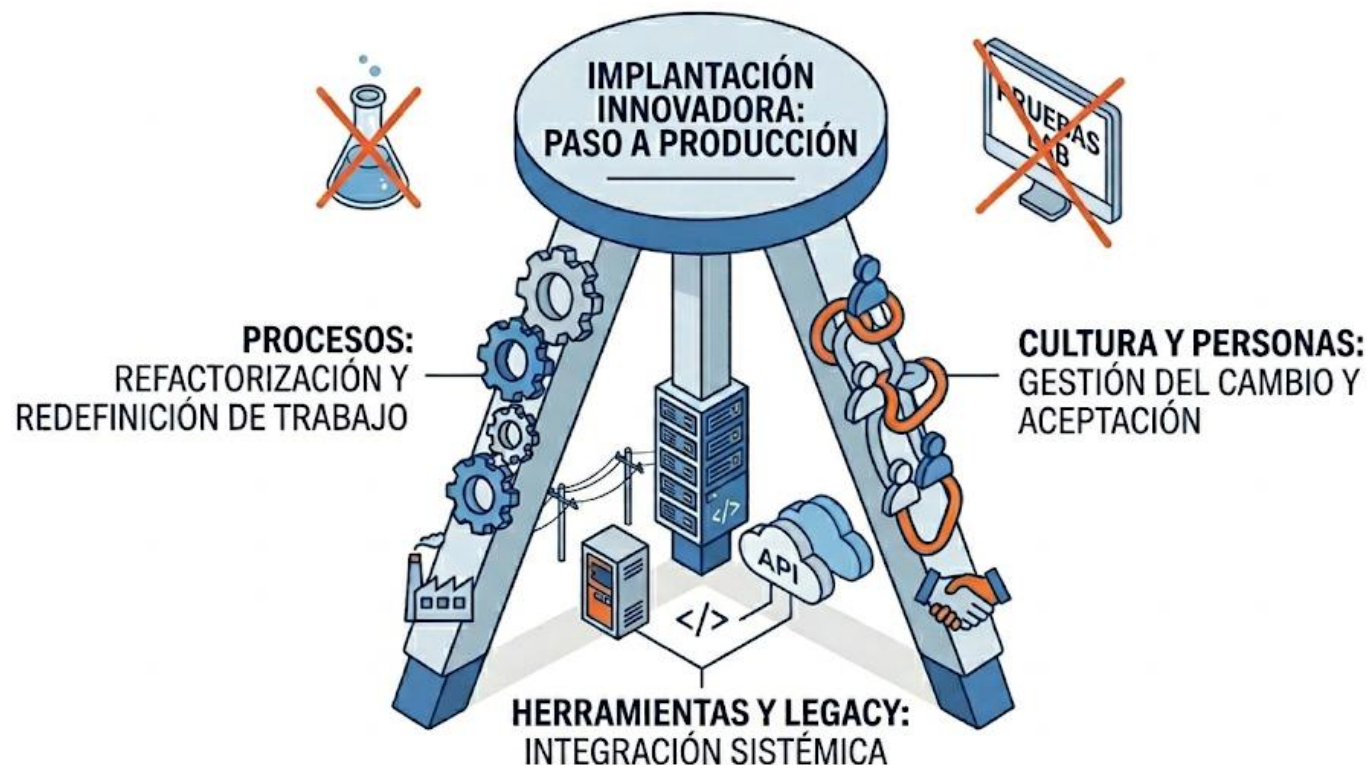
Implantación y escalado

El triángulo de la implantación

Personas, procesos y tecnología

Las tres patas que no se deben olvidar en la mesa de la implantación de cualquier proyecto:

- Herramientas y Legacy (Integración sistémica).
- Procesos (Refactorización).
- Cultura y personas (Gestión del cambio).



Implantación y escalado

Refactorización de procesos

A veces no es solo integrar con lo que hay. Puede que haya que eliminar pasos, adaptar procesos y formas de trabajar.

Hay que conocer los procesos y por qué se hacen cada paso. No vale el "Siempre se ha hecho así".

Puede ser de interés usar herramientas de modelado de procesos como BPM. No es solo dibujar diagramas; es identificar cuellos de botella, redundancias y actividades que no aportan valor.

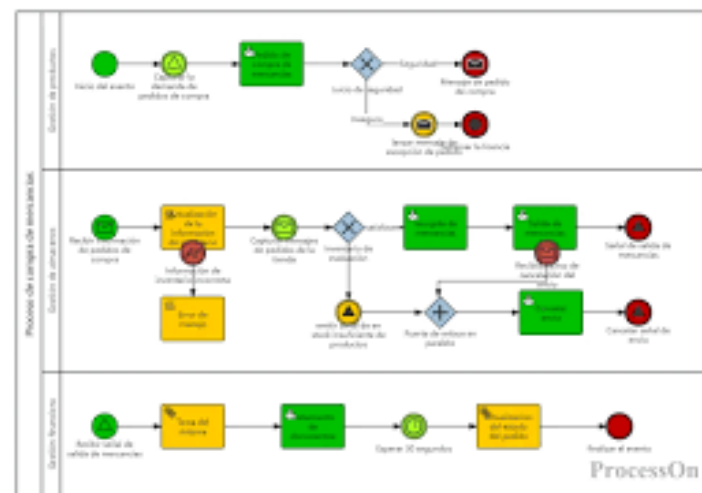
Análisis 'As-Is' vs. 'To-Be'. No puedes diseñar el estado futuro ('To-Be') sin documentar y entender perfectamente el estado actual ('As-Is').



"Automatizar un proceso un inefficiente solo lo hace inefficiente más rápido"

El Riesgo de la automatización ciega: Según datos de consultoras como **McKinsey**, la automatización de un proceso ineficiente sin previo rediseño puede, en el peor de los casos, acelerar la creación de errores o desperdicios.

La clave es: *Simplifica primero, automatiza después.* KISS



Implantación y escalado

Estrategias de "Convivencia" con el legacy

No solo es instalar software, es hablar de **APIs, conectores y limpieza de datos**.

Tener en cuenta la Deuda Técnica.

Introducir una tecnología nueva puede generar costes de mantenimiento futuro si no se integra bien con lo que ya existe.

Para una coexistencia fluida con el software legacy y planificar su eventual sustitución, es fundamental seleccionar una estrategia de integración adecuada.



Estrategias de integración técnica:

- **Arquitectura de Fachada (API First):** No intentes que la nueva tecnología "hable" directamente con la base de datos antigua. Se construye una capa intermedia (API) que moderniza el acceso al sistema legacy.
- **Patrón del "Estrangulador" (Strangler Fig Pattern):** Consiste en ir rodeando el sistema antiguo con funcionalidades nuevas poco a poco, hasta que el sistema legacy queda vacío y puede apagarse. Es la forma más segura de escalar sin detener el negocio.
- **Buses de Datos (Middleware):** Usar herramientas que actúen como "traductores" en tiempo real entre lo viejo y lo nuevo.

Implantación y escalado

La resistencia al cambio

La resistencia al cambio es un reto clave con el que nos vamos a encontrar.

Aquí es donde pasamos de "un experimento" a "la nueva forma de trabajar".

En la gestión del cambio está la diferencia a un "ingeniero de código" de un "ingeniero de soluciones".

Es importante elegir "Evangelistas" internos (empleados que prueban la tecnología primero y convencen a sus compañeros), así como definir planes de formación continua.



Implantación y escalado

Diagnóstico de la resistencia (Modelo ADKAR)

Diagnóstico de la resistencia (Modelo ADKAR)

Este modelo externo es el estándar de oro para entender por qué alguien se resiste. Como ingeniero, debes identificar en qué fase se ha "roto" el usuario:

Awareness (Conciencia):

¿Sabe el usuario *por qué* cambiamos el sistema legacy?

2

Knowledge (Conocimiento): ¿Sabe *cómo* usar la nueva tecnología?

4

Reinforcement (Refuerzo): ¿Se le premia por usar el nuevo sistema o puede volver al viejo sin consecuencias?

1

Desire (Deseo): ¿Ve un beneficio personal ("What's in it for me")?

3

Ability (Capacidad): ¿Tiene el tiempo y apoyo para practicar?

5

Implantación y escalado

Estrategias recomendadas para la gestión del cambio



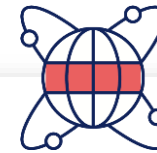
Dimensión Social

- **Evangelistas de la solución (*Champions*):** Identificar *Early Adopters* que influyan en sus pares.
- **Escucha activa y Feedback Loop:** Crear canales para quejas y sugerencias; el usuario debe sentirse escuchado.



El cambio debe valer el esfuerzo:

- **Mejoras evidentes (*Quick Wins*):** Resolver un problema doloroso en las primeras semanas.
- **Transparencia en el "Por qué":** Comunicar el propósito vital de la herramienta para la supervivencia de la empresa.



Para integrar la tecnología sin fricción:

- **Exposición gradual (*Phased Rollout*):** Despliegue paulatino (ej. con *feature toggles* o por departamentos menos críticos).
- **Diseño empático (UX/UI):** Interfaz simple y transparente centrada en el operario, no en el ingeniero. Menos clics, menos resistencia.
- **Sistemas de recompensa (*Gamificación*):** Uso de mecánicas de juego (ej. barras de progreso) para incentivar el uso correcto.



"Lo que seguro que no funciona: Manuales en PDF de 50 páginas"

Implantación y escalado

El día después

Hay algunos factores que debemos tener presentes, algunos desde el inicio y en todo el proceso, pero otros entran en marcha el día después.

Debemos evitar el “síndrome de parto”: nos obsesionamos con el lanzamiento y nos olvidamos de la crianza.

- Modelo de gobierno.
- Los costes de mantenimiento.
- El escalado.
- Partners y colaboradores.
- La seguridad por defecto.
- Observabilidad.



Implantación y escalado

Modelo de gobierno (Governance)

Sería la tutela de nuestro recién nacido.

No se trata de mandar, sino de establecer las reglas del juego. Según COBIT, el gobierno asegura que la tecnología entregue valor y los riesgos se gestionen.



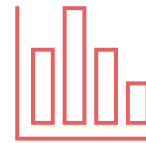
Toma de decisiones

¿Quién decide cuándo se actualiza la tecnología emergente? Debe haber un comité que alinee IT y Negocio.



Políticas de uso

Establecer marcos éticos y legales (ej. si usamos IA, ¿quién es responsable de sus errores?).



Métricas de valor (KPIs)

Pasar de medir "si el servidor está encendido" a medir "cuánto valor de negocio aporta la solución mensualmente".

Implantación y escalado

Los costes de mantenimiento (TCO)

El sustento

El "hijo" necesita recursos constantes

Muchos ingenieros solo ven el coste de desarrollo (CAPEX), pero el 70% del coste real de una tecnología ocurre después del despliegue (OPEX).

- Deuda Técnica: si usamos una tecnología muy nueva (TRL bajo), el coste de encontrar desarrolladores que la mantengan será mayor.
- Licenciamiento y Cloud: el escalado de soluciones emergentes en la nube puede disparar los costes si no hay un control de FinOps.
- Evolución obligatoria: las tecnologías emergentes cambian rápido; mantener la compatibilidad con el sistema legacy requiere inversión constante.



Implantación y escalado

El escalado (Scalability & Elasticity)

El crecimiento

Debemos preparar la infraestructura para cuando el éxito del proyecto demande más capacidad.

Hay que prever este crecimiento y la demanda que esto va a suponer.

El reto aquí es pasar de 100 usuarios (piloto) a 10.000 (producción).

- Escalado horizontal vs. vertical: diseñar la solución para que pueda crecer sin rediseñar el código.
- Rendimiento bajo carga: monitorizar la latencia del sistema legacy cuando la nueva tecnología le pide miles de datos por segundo.
- Automatización (IaC): usar Infraestructura como Código para que el escalado sea repetible y no dependa de intervenciones manuales.



Implantación y escalado

Partners y colaboradores

La red de apoyo

Nadie cría a un hijo solo. Dependemos de proveedores de servicios y comunidades.

Al seleccionar una tecnología emergente, es crucial considerar el ecosistema de partners y colaboradores con el que contamos.

En tecnologías emergentes, nadie camina solo. La dependencia de terceros es un riesgo estratégico.

- Vendor Lock-in: evaluar si estamos tan atados a un proveedor (ej. OpenAI o AWS) que no podríamos cambiarnos si suben los precios o cierran.
- Soporte y comunidad: si el partner desaparece, ¿existe una comunidad de código abierto que mantenga la tecnología?
- SLA de terceros: los acuerdos de nivel de servicio de nuestros partners deben estar alineados con los que nosotros prometemos al negocio.



Implantación y escalado

La seguridad por defecto (Security by Design)

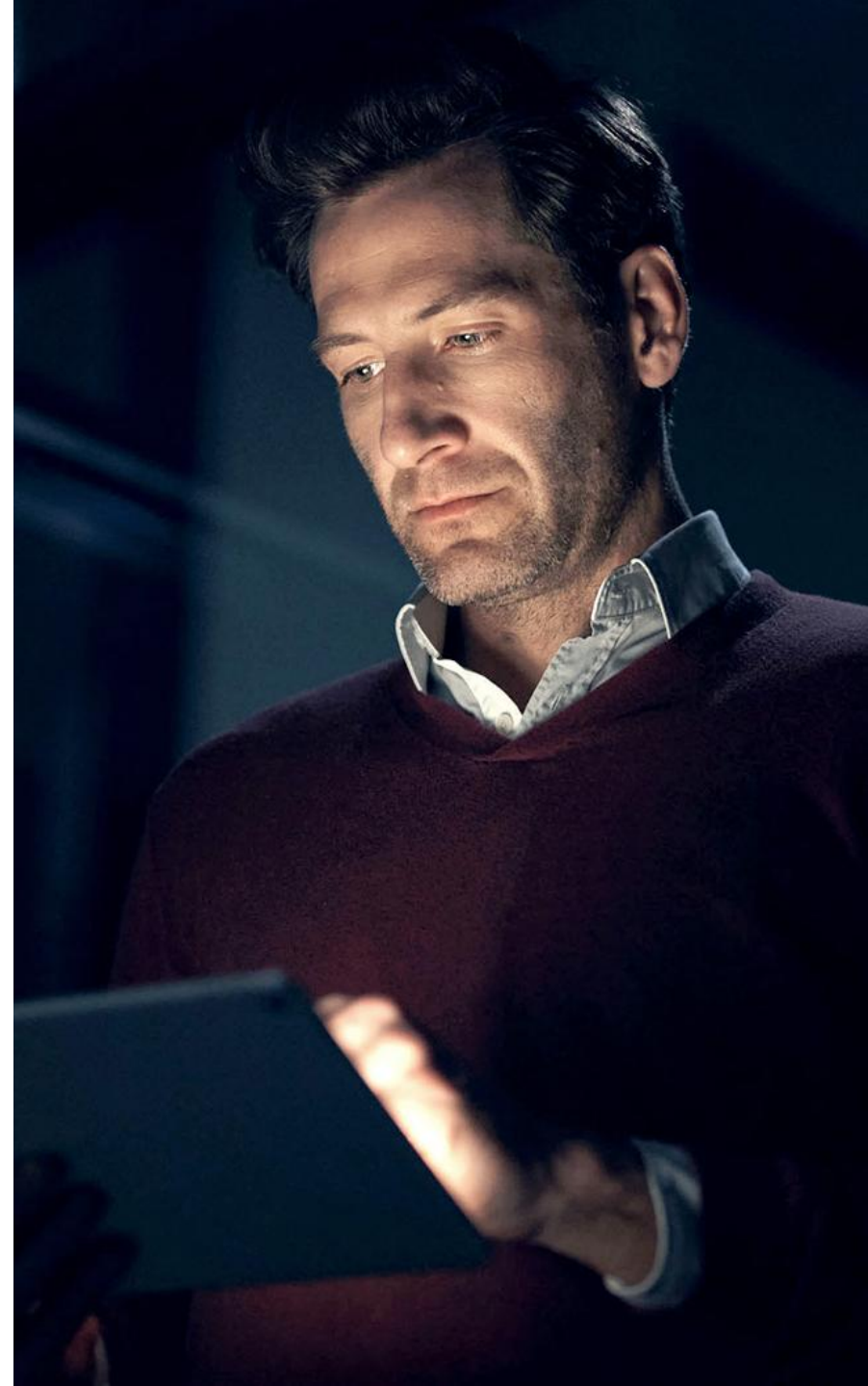
La protección

La seguridad no es un evento, es un estado constante de vigilancia.

Contrastando con el marco NIST, la seguridad no es un parche final, sino una propiedad intrínseca que ahora debe ser auditada en producción.

De hecho, uno de los puntos que debemos de tener en cuenta en todo el proceso desde el diseño hasta la puesta en producción y explicación del sistema.

- Gestión de vulnerabilidades: las tecnologías nuevas suelen tener agujeros de seguridad no descubiertos. Necesitamos escaneos constantes.
- Privacidad de los datos: asegurar que la integración con el sistema antiguo no expone datos sensibles (GDPR).
- Seguridad en la integración: el "puente" que construimos entre lo nuevo y lo viejo suele ser el punto más débil para un ciberataque.



Implantación y escalado

Observabilidad y feedback

El chequeo médico

Es el sistema sensorial que nos dice si la tecnología está sana y si cumple su propósito.

- Telemetría: medir latencias, errores y consumo de recursos en tiempo real.
- Voz del usuario: ¿la herramienta está ayudando o es un estorbo? El feedback del usuario nos proporciona información importante a tener en cuenta a que es el alimento de la mejora continua o llamadas de atención.



Actividad. El forense de la implantación (opcional)

Objetivo: identificar por qué una tecnología emergente se estanca o fracasa durante su despliegue real.



02

Herramientas y metodologías para la innovación



02.1

Introducción



Introducción

La innovación como proceso



Las tecnologías emergentes suelen integrarse usando metodologías de innovación y desarrollo iterativo.

En este bloque formativo, exploraremos cómo las organizaciones no solo generan ideas, sino cómo las transforman en productos reales mediante un marco metodológico estructurado.

El objetivo es entender que la innovación no es un evento fortuito, sino un proceso iterativo y adaptable.

Introducción

Hoja de ruta de la unidad 2

A lo largo de la sesión, desglosaremos las herramientas líderes en la industria siguiendo este itinerario:

01

Identificación y empatía (Design Thinking)

Veremos cómo detectar problemas reales centrados en el usuario antes de empezar a programar o desarrollar.

02

Validación de hipótesis (Lean Startup)

Aprenderemos a utilizar el Producto Mínimo Viable (MVP) para validar tecnologías en el mercado de forma rápida.

03

Ejecución y agilidad (Agile / Scrum)

Analizaremos el marco de trabajo más utilizado para gestionar la incertidumbre y realizar entregas frecuentes de valor.

Introducción

Nuevo marco conceptual

La gestión de proyectos moderna requiere adaptabilidad frente al cambio acelerado. La aplicación de tecnologías emergentes demanda una filosofía que priorice:

Procesos iterativos y flexibles

Ante el contexto VUCA, los proyectos deben usar enfoques ágiles (Scrum, Kanban) con ciclos cortos para entregar valor y permitir la adaptación rápida a nueva información y riesgos. La tecnología emergente facilita y requiere esta agilidad.

Diseño Centrado en el Ser Humano (Human-Centric Design)

El éxito se mide por la satisfacción del usuario. Las tecnologías deben resolver problemas humanos significativos. Es vital integrar el feedback de los stakeholders para asegurar que la innovación tecnológica aporte valor tangible.

Creatividad, Experimentación y Aprendizaje Continuo

Para aprovechar el potencial de tecnologías como IA, Blockchain o IoT, la cultura del proyecto debe fomentar la experimentación controlada y aceptar el fracaso como fuente de aprendizaje, impulsando soluciones disruptivas e innovadoras.

02.2

Design Thinking: Más allá del código



Design Thinking: Más allá del código

Design Thinking es una metodología de resolución de problemas centrada en la persona, que permite identificar oportunidades donde la tecnología realmente aporta valor.

Antes de la primera línea de código, está la necesidad humana.

Las 5 Estaciones del Proceso:

1. **Empatizar:** Salir de la terminal y observar al usuario.
2. **Definir:** Detectar el "punto de dolor" (Pain Point) real.
3. **Idear:** Pensar sin las limitaciones del sistema actual.
4. **Prototipar:** Construir para aprender, no para producción.
5. **Testear:** Validar con el usuario para iterar rápido.



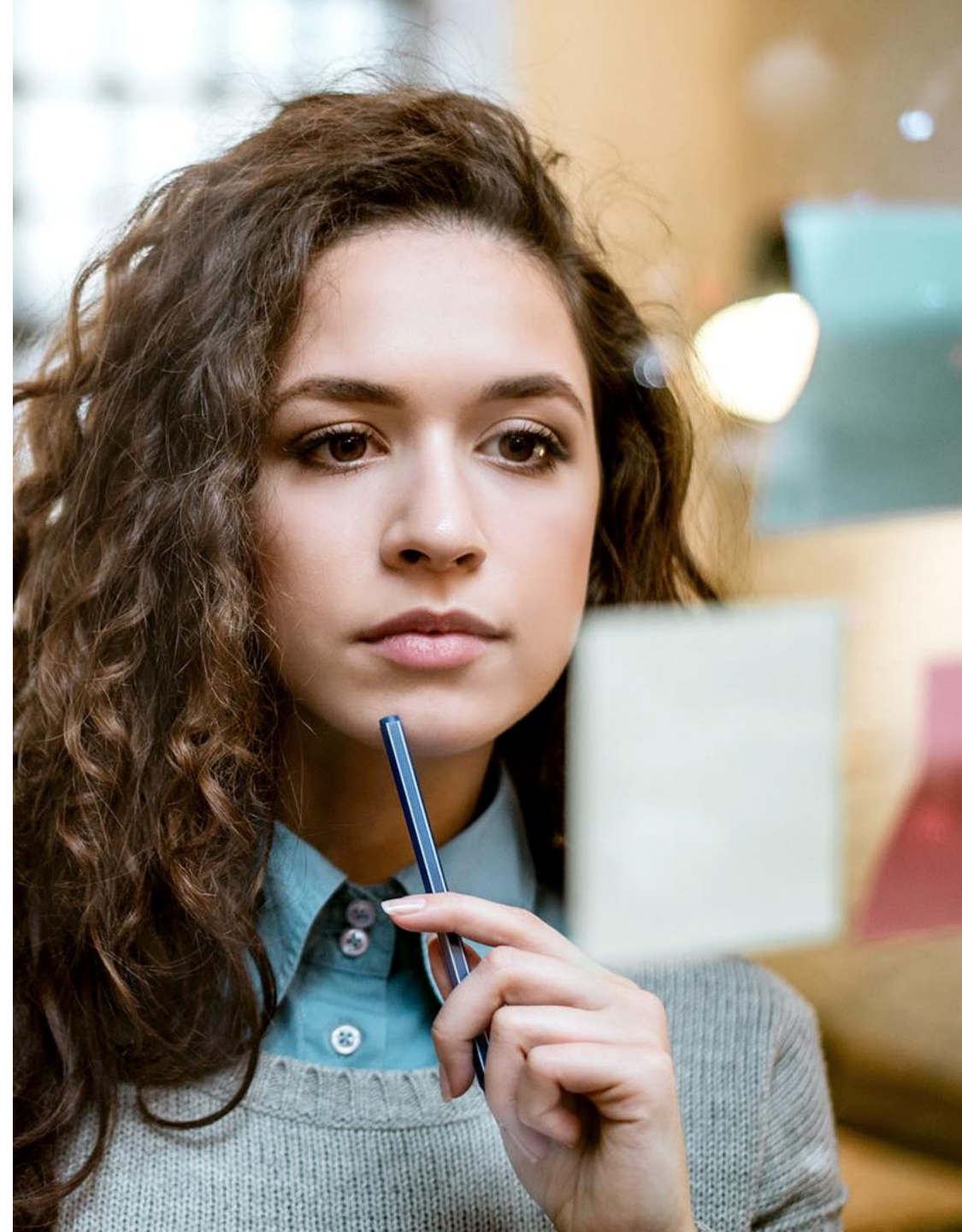
“El Design Thinking es el filtro que evita que construyamos catedrales tecnológicas que nadie quiere visitar.”

Design Thinking: Más allá del código

Entender el problema antes de escribir la solución

La ingeniería al servicio de la necesidad real.

- **Objetivo:** Pasar del "¿Puedo construirlo?" al "¿Debo construirlo?".
- **Evitar la "Paradoja del Martillo":** No permitas que la tecnología (IA, IoT, etc.) condicione el análisis del problema.
- **Foco en el valor:** Si la solución técnica no resuelve un "punto de dolor" real, no es innovación, es un gasto.
- **La regla de oro:** Invertir en entender el contexto del usuario reduce el riesgo de rechazo tecnológico.



Design Thinking: Más allá del código

“Empatizar” y “Definir”. Problem Space

Empatizar

Inmersión en el contexto.

- Observación directa y entrevistas.
- Detección de "soluciones de parche" y necesidades reales.
- Herramienta: Mapa de Empatía.

Definir

Síntesis y foco.

- Filtrar el ruido para encontrar el Insight (la clave del problema).
- Construcción del POV (Point of View).
- Ecuación: Usuario + Necesidad + Insight.

El reto

No busques la solución todavía. Asegura que estás resolviendo el problema correcto.

Design Thinking: Más allá del código

"Idear" y "Prototipar". Solution Space

Construir para aprender, no para producir.



Idear

Creatividad sin filtros.

- Generar soluciones que aprovechen las tecnologías emergentes.
- Técnicas: Brainstorming y SCAMPER.
- Objetivo: Divergir para encontrar la solución más disruptiva.



Prototipar

Hacer las ideas tangibles.

- Construcción de modelos de baja fidelidad (rápidos y baratos).
- Foco en la interacción, no en la robustez técnica.
- Herramienta: Mockups, Figma, Prototipos de papel o "Mago de Oz".



Consejos

- Si un prototipo tarda más de un par de días en construirse, estás desarrollando, no prototipando.
- Falla rápido y barato: El código es caro, el papel es barato.

Design Thinking: Más allá del código

Testear: Validar para Iterar

No diseñamos para nosotros, sino para ellos.

- La Prueba: Poner el prototipo en un escenario real con usuarios reales.
- Escucha Activa: Evitar vender la solución.
 - Observar el comportamiento, no solo escuchar la opinión.
- El Resultado:
 - Validar: La idea funciona, seguimos hacia el MVP.
 - Iterar: Aprender del error y volver a una fase anterior (Empatizar, Definir o Idear).



"Falla rápido para tener éxito pronto"



Design Thinking: Más allá del código

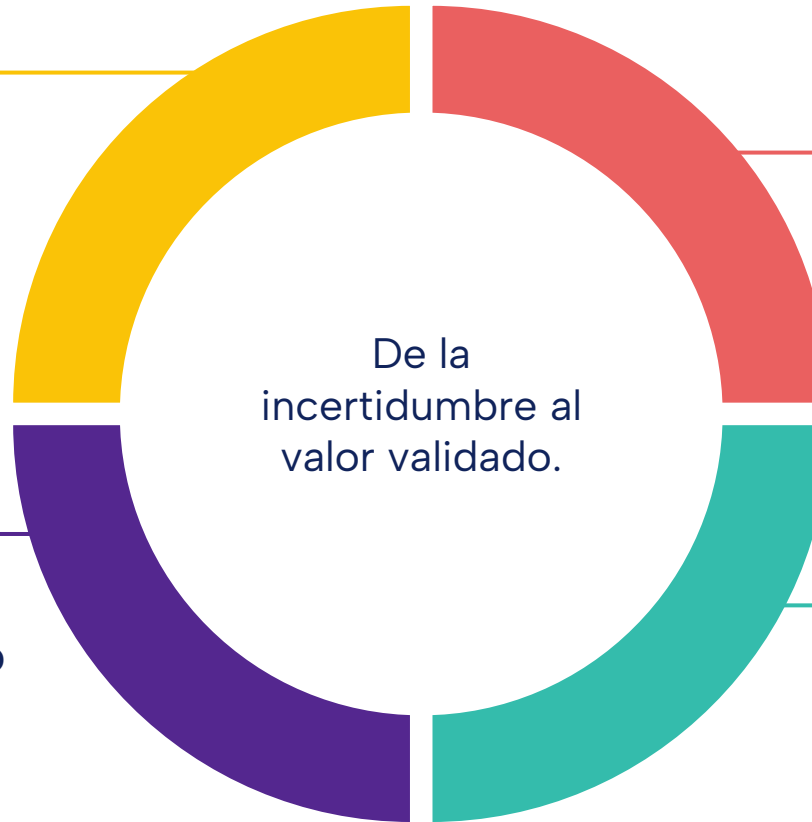
Conclusiones de Design Thinking

Aprendizaje validado:

hemos sustituido opiniones por evidencias del usuario.

Próximo paso:

¿cómo transformamos este prototipo en un negocio o producto sostenible? → Lean Startup.



Reducción de riesgo: es más barato fallar en un prototipo que en un despliegue de producción.

Foco en el impacto: la tecnología emergente ahora tiene un propósito claro y un usuario que la espera.

Actividad.

El Ingeniero empático

Objetivo: aplicar las dos primeras fases del Design Thinking (**Empatizar y Definir**) para descubrir necesidades reales en la Residencia Horizonte que no son evidentes a simple vista.



Actividad. El Hackatón SCAMPER

(opcional)

¿Qué significa SCAMPER? (Aplicado a ingeniería)

Cada letra es una acción que debes aplicar a tu producto o proceso actual:

- **S – Sustituir:** ¿Qué componente, tecnología o material podemos cambiar?
- **C – Combinar:** ¿Podemos mezclar dos funciones o servicios en uno solo?
- **A – Adaptar:** ¿Hay alguna solución en otro sector que nos sirva?
- **M – Modificar / Magnificar:** ¿Qué pasa si cambiamos la escala, el tamaño o la velocidad?
- **P – Poner en otros usos:** ¿Podemos usar esta tecnología para algo totalmente distinto?
- **E – Eliminar:** ¿Qué sobra? ¿Qué podemos simplificar al máximo? (Casi siempre es la más valiosa para un ingeniero).
- **R – Reorganizar / Revertir:** ¿Qué pasa si cambiamos el orden del proceso o el diseño?



02.3

Lean Startup

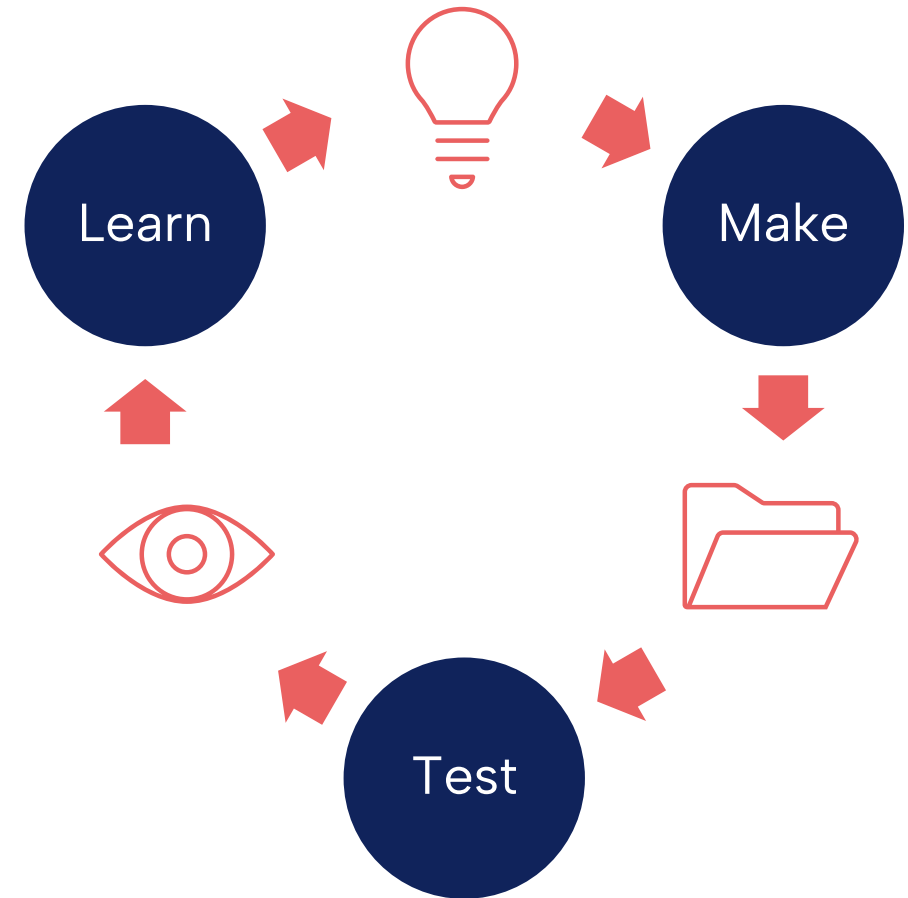


Lean Startup

Conclusiones de Design Thinking

Lean Startup: Ingeniería basada en el aprendizaje.

- Origen: Eric Ries (Silicon Valley, post-burbuja puntocom).
- Influencia Lean: Inspirado en el sistema de producción de Toyota (Eliminación de desperdicio).
- El cambio de paradigma:
 - Antes: planificación rígida y ejecución larga.
 - Ahora: experimentación científica e iteración rápida.
- Unidad de Progreso: el Aprendizaje Validado (no las líneas de código).
- El ciclo fundamental: Crear – Medir – Aprender.



Lean Startup

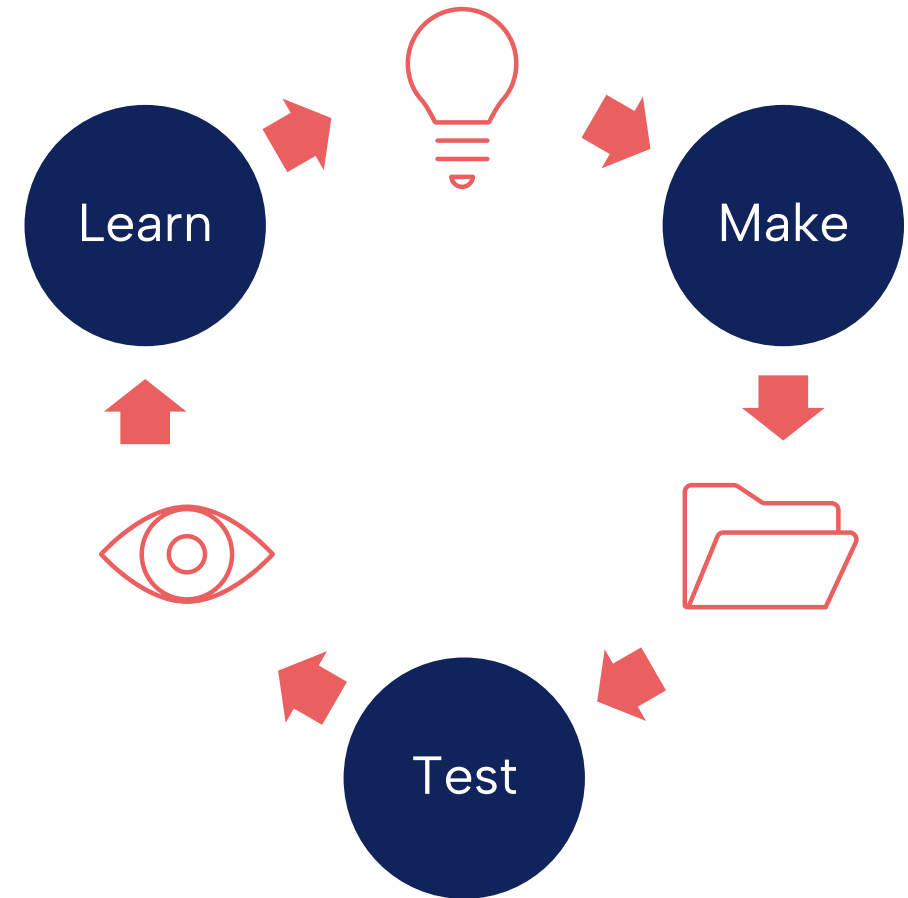
El ciclo de aprendizaje validado

Convertir ideas en datos y datos en decisiones

- **Crear:** crear el MVP para poner a prueba nuestra hipótesis técnica.
- **Medir:** obtener métricas reales de uso. ¿Cómo reacciona el sistema y el usuario?
- **Aprender:** analizar si debemos **Pivotar** (cambiar de rumbo) o **Perseverar** (mejorar lo existente).

META: minimizar el tiempo total a través del ciclo.

Mantra: "no fracasas si aprendes algo que te permite ahorrar recursos en el futuro".

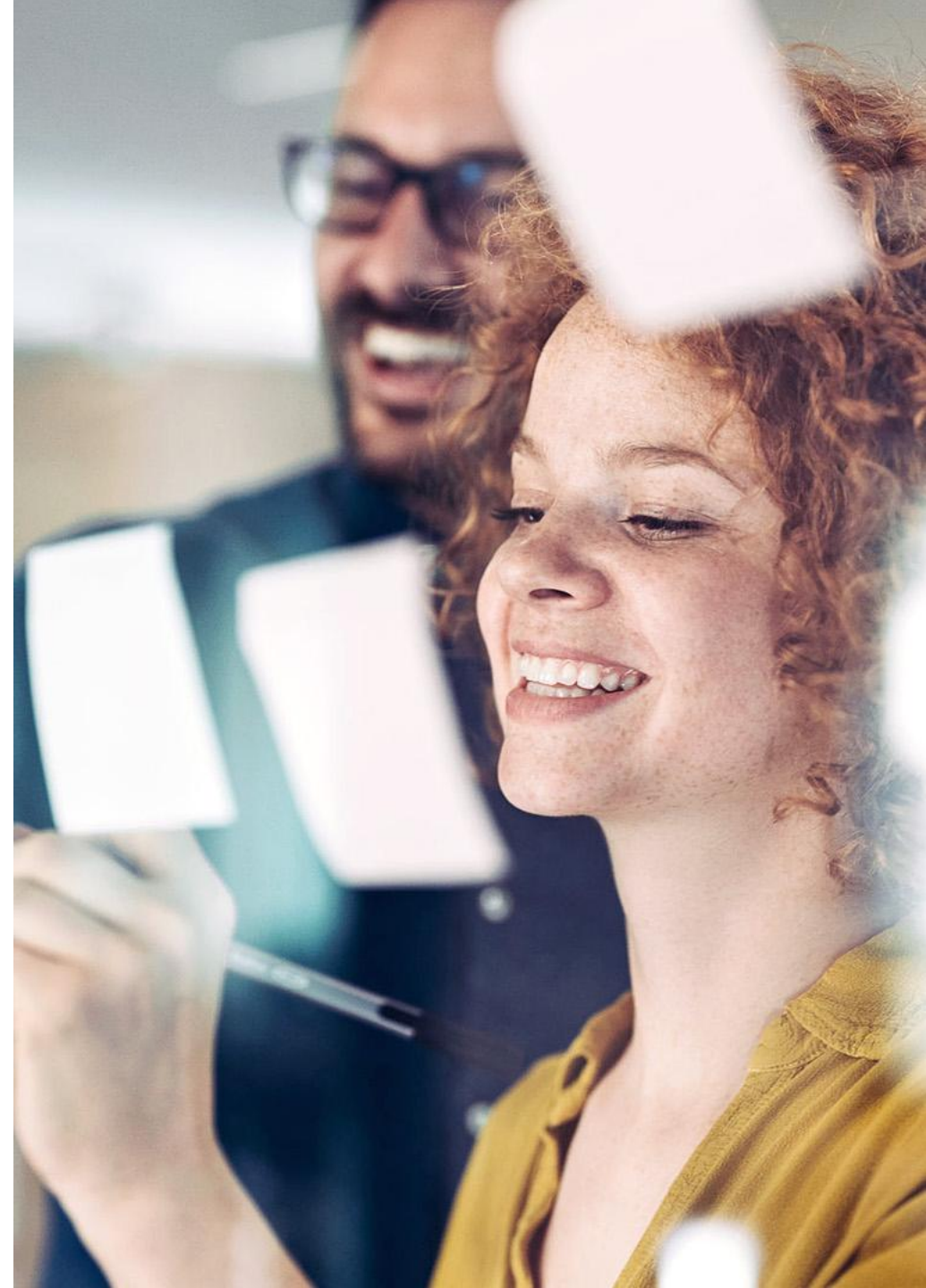


Lean Startup

Hipótesis validadas por método científico

Lean Startup: Ingeniería basada en el aprendizaje.

- Hipótesis de valor: ¿El producto soluciona algo realmente?
- Hipótesis de crecimiento: ¿Cómo se descubrirá y escalará el producto?
- Transformar el "creemos que..." en "si hacemos X, entonces pasará Y".



Lean Startup

El MVP (Producto Mínimo Viable)

El MVP no es un producto "a medias"

Objetivo: Obtener datos reales con el mínimo esfuerzo técnico posible.

- Versión de un nuevo producto que permite recolectar el máximo aprendizaje validado con el mínimo esfuerzo.
- No es un prototipo desechable, es una herramienta de aprendizaje.
- Mínimo: Lo más pequeño posible.
- Viable: Que aporte valor real.
- Mago de Oz: Interfaz funcional, proceso manual detrás. (Parece IA, es humano).
- Conserje: Servicio manual de alto valor para entender al usuario.
- Humo (Landing Page): Medir el interés real mediante clics o registros antes de construir.
- Prototipo en video: Mostrar cómo funcionaría la tecnología (caso Dropbox).

Lean Startup

Métricas que importan

Medir: Evitando las "Métricas de vanidad"

Métricas de vanidad: Descargas totales, visitas, likes (te hacen sentir bien, pero no informan).

Métricas accionables:

- Retención: ¿vuelven los usuarios?
- Activación: ¿prueban la funcionalidad estrella (IA, IoT)?
- Engagement: frecuencia de uso real.

Contabilidad de la innovación. Aprendizaje validado.



Lean Startup

El momento de la verdad: Pivotar o perseverar

Aprender: ¿Seguimos o cambiamos el rumbo?

- Pivotar: Cambio de estrategia sin cambiar la visión (cambiar el segmento de cliente, el modelo de ingresos, etc.). Golpe de timón.
- Perseverar: Los datos validan la hipótesis. Iteramos para optimizar y escalar.
- El "Fracaso" es Aprendizaje Validado:
 - Fallo detectado = Recurso ahorrado.
 - No es un error del equipo, es una validación negativa del mercado.



Lean Startup

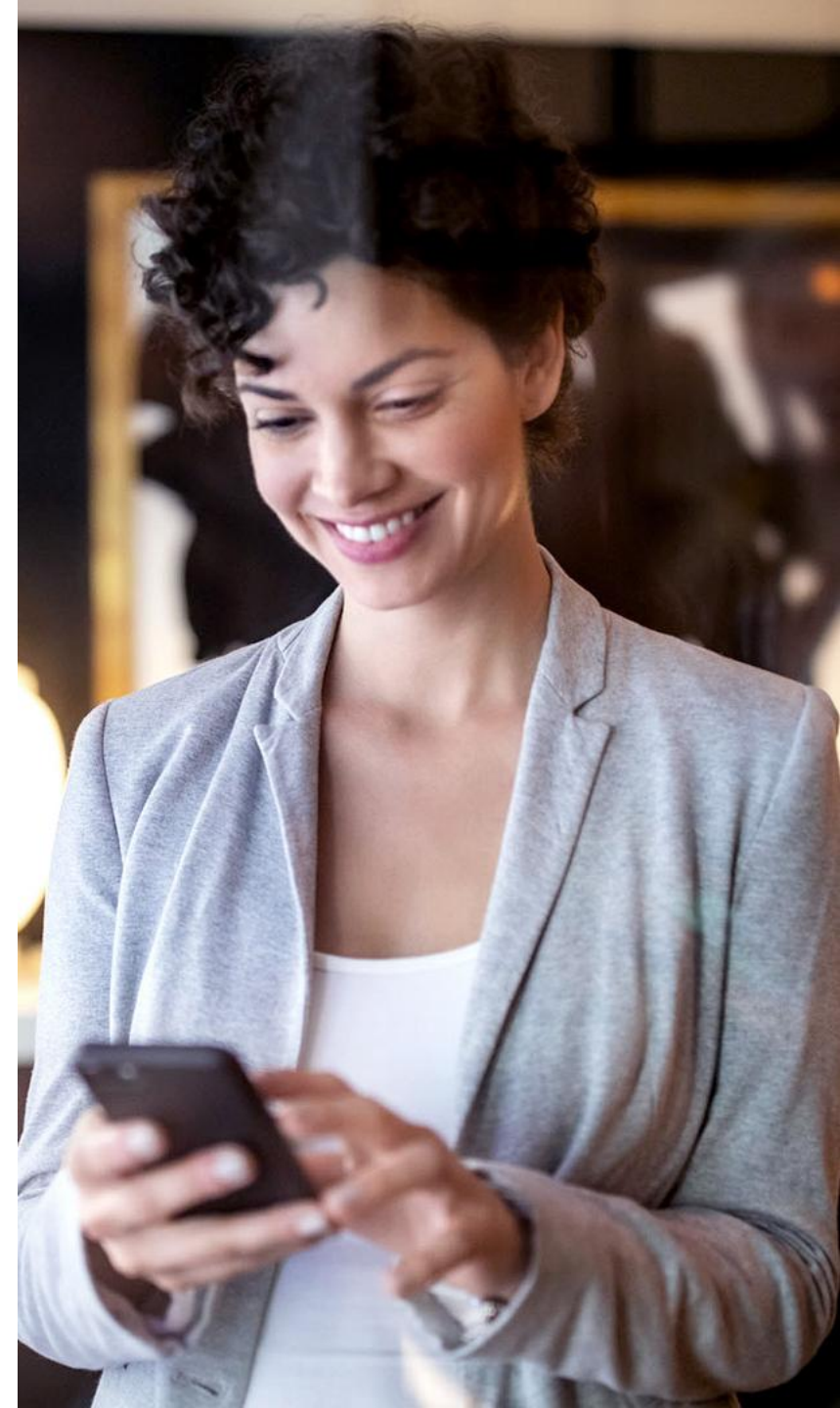
El flujo de la innovación: “Entender”, “Validar” y “Construir”

Seguimos el proceso de la innovación

- Design Thinking: Explorar el problema y empatizar (Hacer las cosas correctas).
- **Lean Startup: Validar la solución y el mercado (Hacer lo que funciona).**
- Metodologías de desarrollo ágil: Ejecutar y entregar con calidad (Hacerlo de la forma correcta).



Resultado: reducción drástica del riesgo técnico y de negocio.



Actividad.

El laboratorio del MVP

Objetivo: diseñar un experimento para validar la hipótesis principal de un proyecto sin escribir una sola línea de código.



02.4

Desarrollo con
metodologías ágiles:
SCRUM



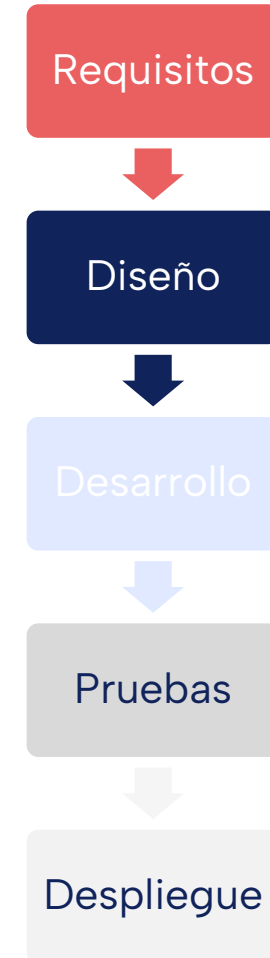
Desarrollo con metodologías ágiles: SCRUM

- Modelo Waterfall o en cascada: Antes de los años 90, el software se desarrollaba igual que se construye un puente o un edificio.
- **Crisis del Software:** Proyectos fuera de plazo, coste y sin valor real.
- **El Hito:** El Manifiesto Ágil (2001).



“Las metodologías ágiles, no aparecieron por moda, sino como una rebelión técnica contra un sistema que no funcionaba en el desarrollo de software”.

Modelo Waterfall



Desarrollo con metodologías ágiles: SCRUM

De la teoría a la entrega

Se formalizó con el **Manifiesto Ágil de 2001 con 4 valores:**

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Respuesta ante el cambio sobre seguir un plan.



<https://agilemanifesto.org/iso/es/manifesto.html>



"Ser ágil no es "ir más rápido", es "entregar valor antes" y tener la capacidad de maniobra que el mundo tecnológico actual exige".

Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- **Individuos e interacciones** sobre procesos y herramientas
- **Software funcionando** sobre documentación extensiva
- **Colaboración con el cliente** sobre negociación contractual
- **Respuesta ante el cambio** sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Kent Beck
Mike Beedle
Arie van
Bennekum
Alistair
Cockburn
Ward
Cunningham

Martin Fowler
James
Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C.
Martin
Steve Mellor
Ken Schwaber
Jeff
Sutherland
Dave Thomas

Desarrollo con metodologías ágiles: SCRUM

¿Por qué para tecnologías emergentes?

Con tecnologías emergentes aumenta la incertidumbre.

La agilidad permite fallar en un Sprint de 2 semanas en lugar de fallar en un proyecto de 2 años.

Principios de metodologías ágiles:

- Desarrollo iterativo.
- Adaptación al cambio.
- Colaboración con el cliente.



No confundir "ser ágil" con "hacer las cosas rápido y mal" (el famoso *cowboy coding*).



Desarrollo con metodologías ágiles: SCRUM

SCRUM

- Marco de trabajo ágil más utilizado.
- Basado en la transparencia, inspección y adaptación.
- Foco en software funcional, no en documentación exhaustiva.



Desarrollo con metodologías ágiles: SCRUM

SCRUM

Scrum Team

- Product Owner
- Developers
- Scrum Master

Artifacts

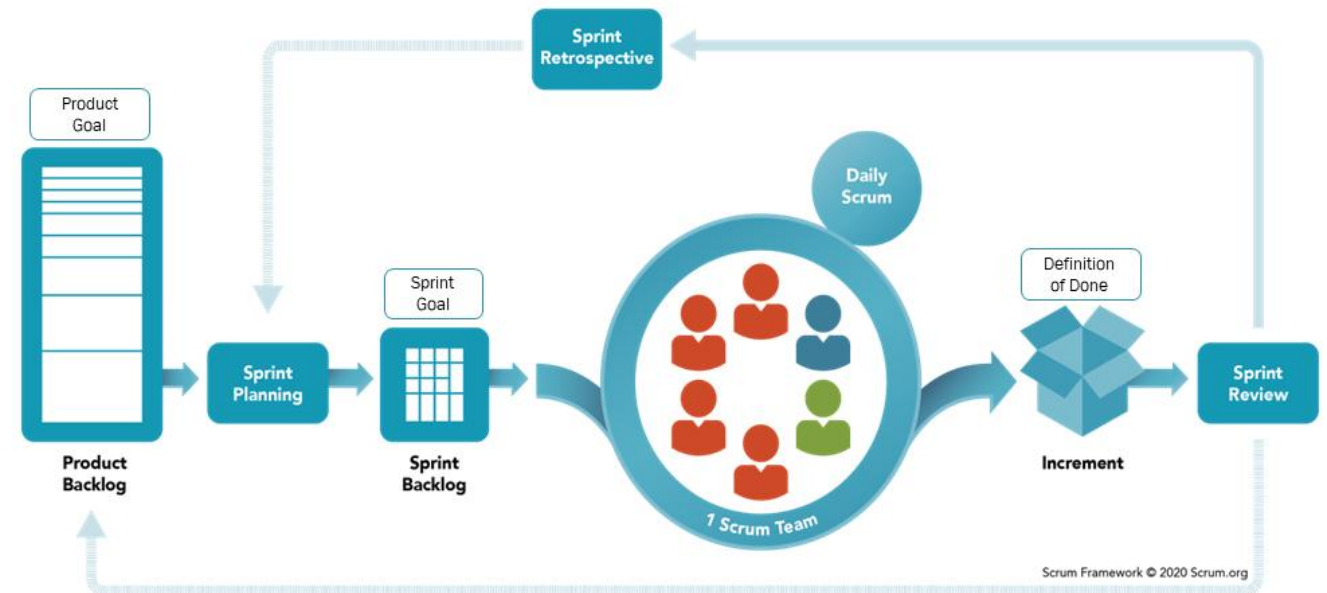
- Product Backlog
- Sprint Backlog
- Increment

Commitments

- Product Goal
- Sprint Goal
- Definition of Done

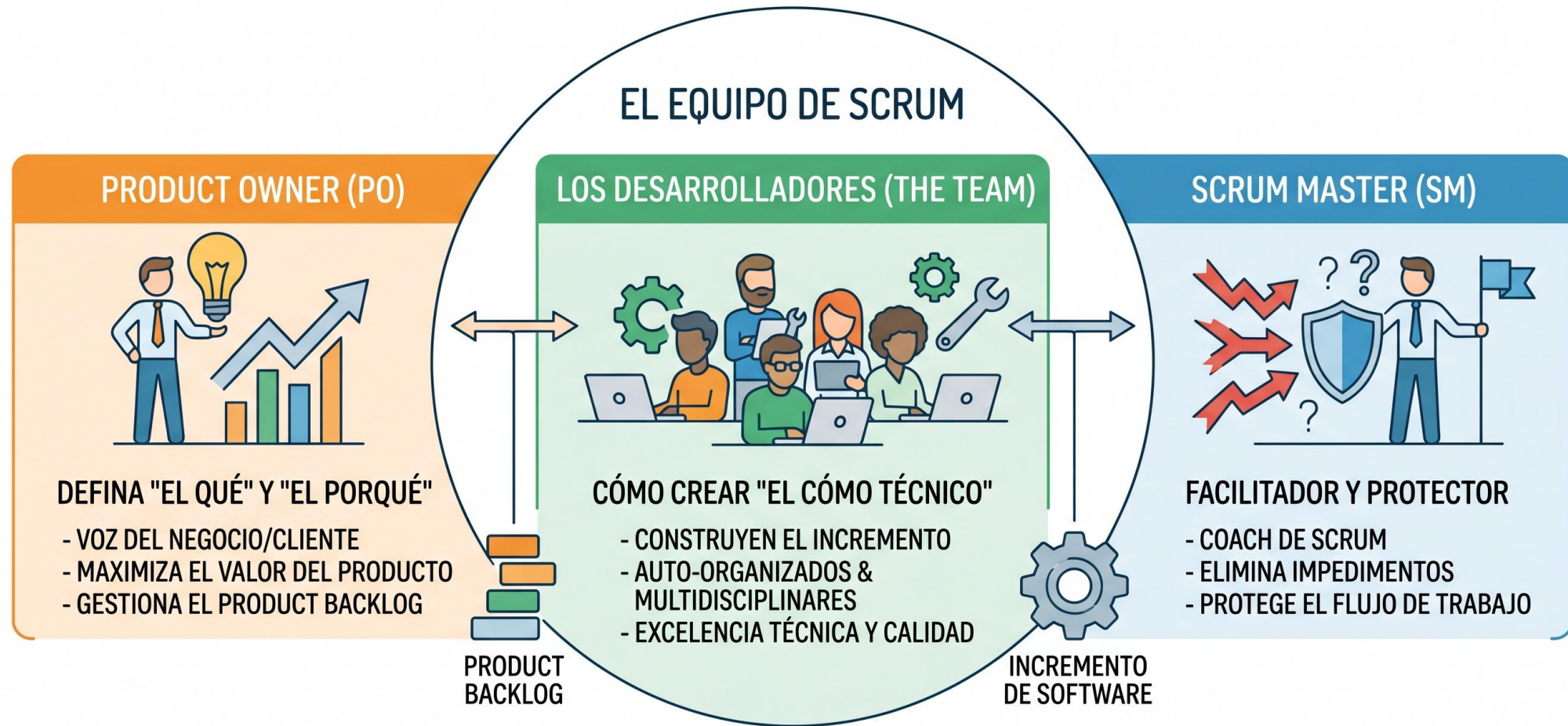
Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective



Desarrollo con metodologías ágiles: SCRUM

Los roles de SCRUM: el SCRUM team



Desarrollo con metodologías ágiles: SCRUM

Artefactos: Visibilidad de principio a fin.

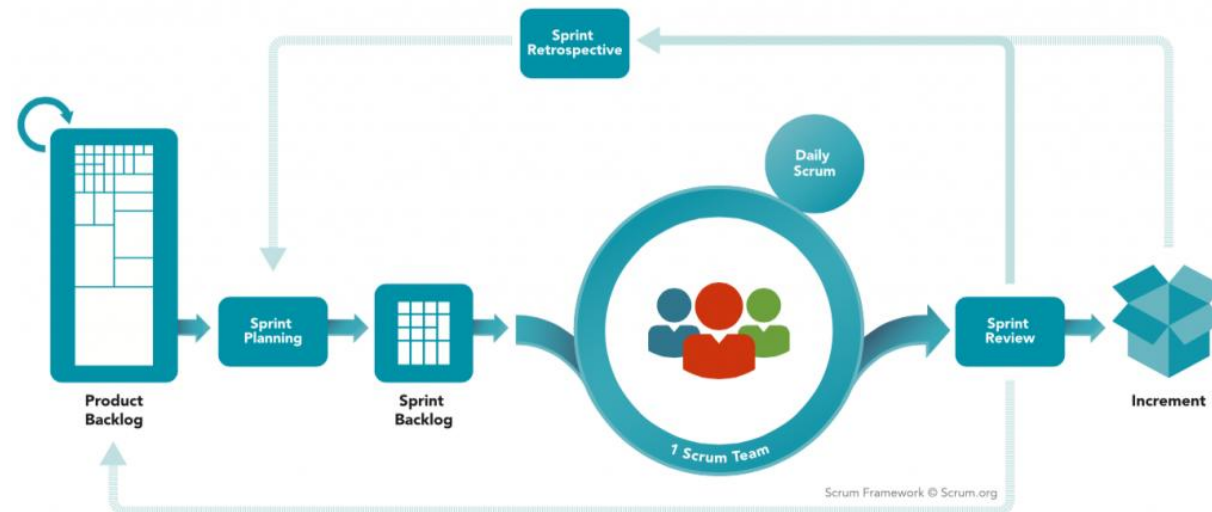
Product Backlog:

- Lista única y ordenada de funcionalidades (User Stories), mejoras y correcciones.
- Flexible y priorizado por valor de negocio.
- Incremento: El software "terminado" y listo para producción.

Sprint Backlog:

El plan técnico para las próximas 2-4 semanas.

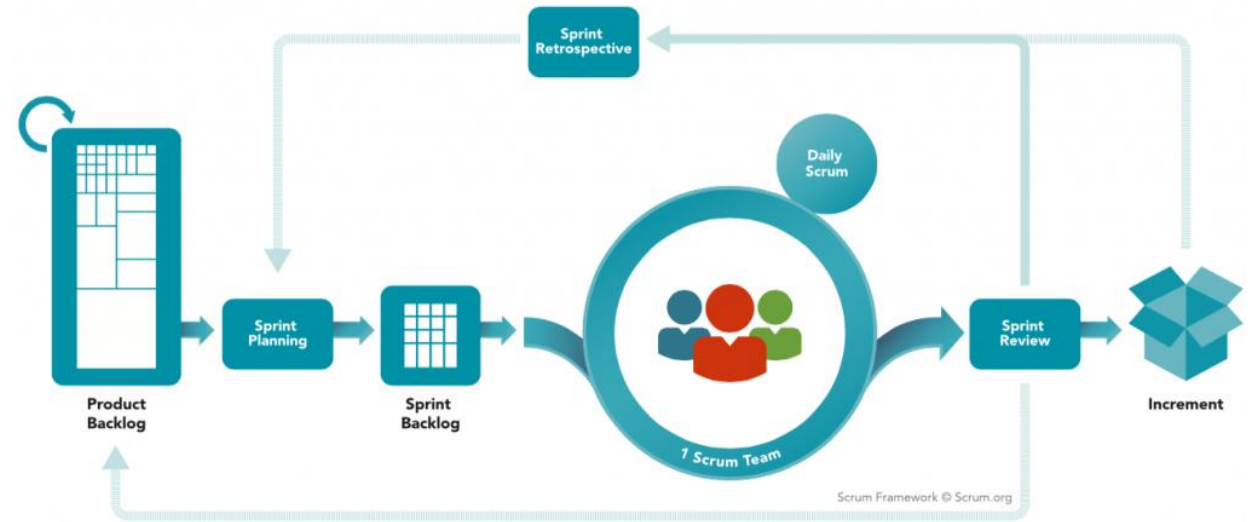
- Selección de tareas para el Sprint y plan de ejecución.
- Propiedad exclusiva de los Desarrolladores.



Desarrollo con metodologías ágiles: SCRUM

El Sprint: el motor de la iteración

Sprint Backlog (SB): Es el subconjunto del PB que el equipo selecciona para el Sprint actual, junto con el plan para entregarlo. Un contenedor para todos las HU (frecuentemente 2 semanas).



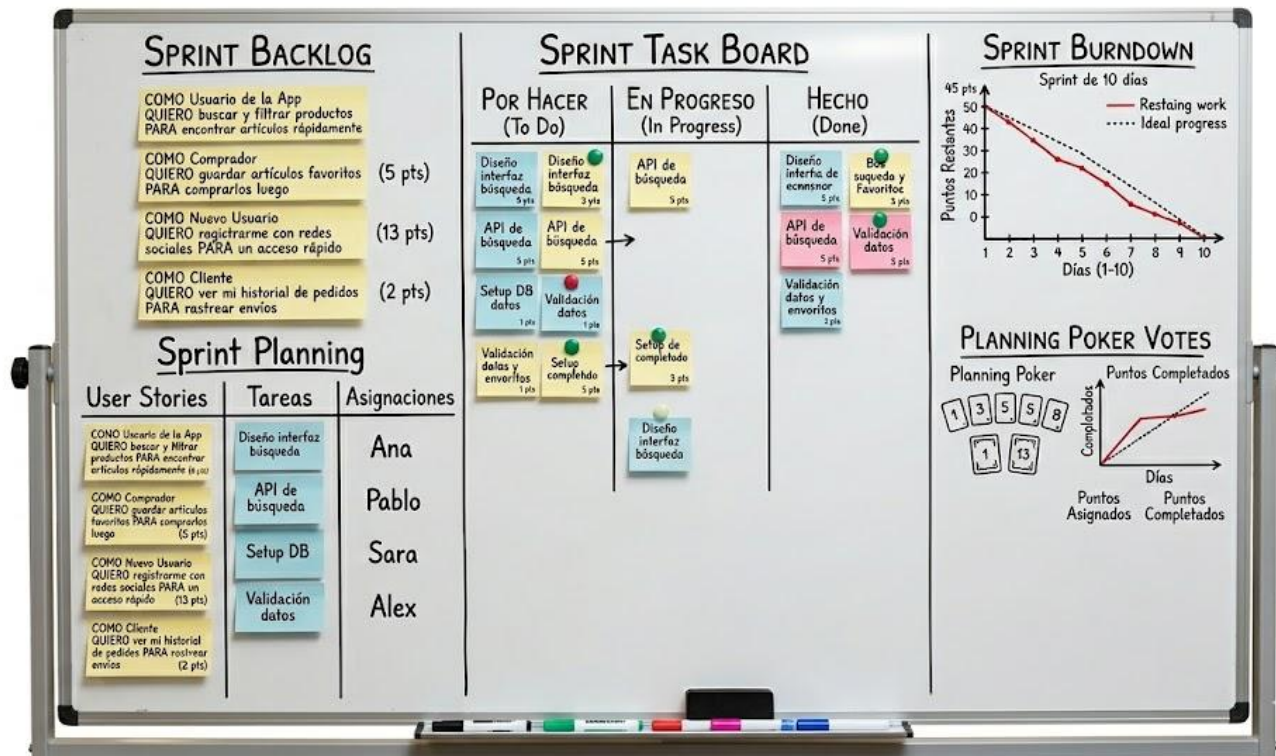
Sprint Planning:
Definición del
objetivo del Sprint.

Daily Scrum:
Sincronización
técnica diaria de
15 min.



Desarrollo con metodologías ágiles: SCRUM

Más artefactos: Herramientas de estimación y gestión

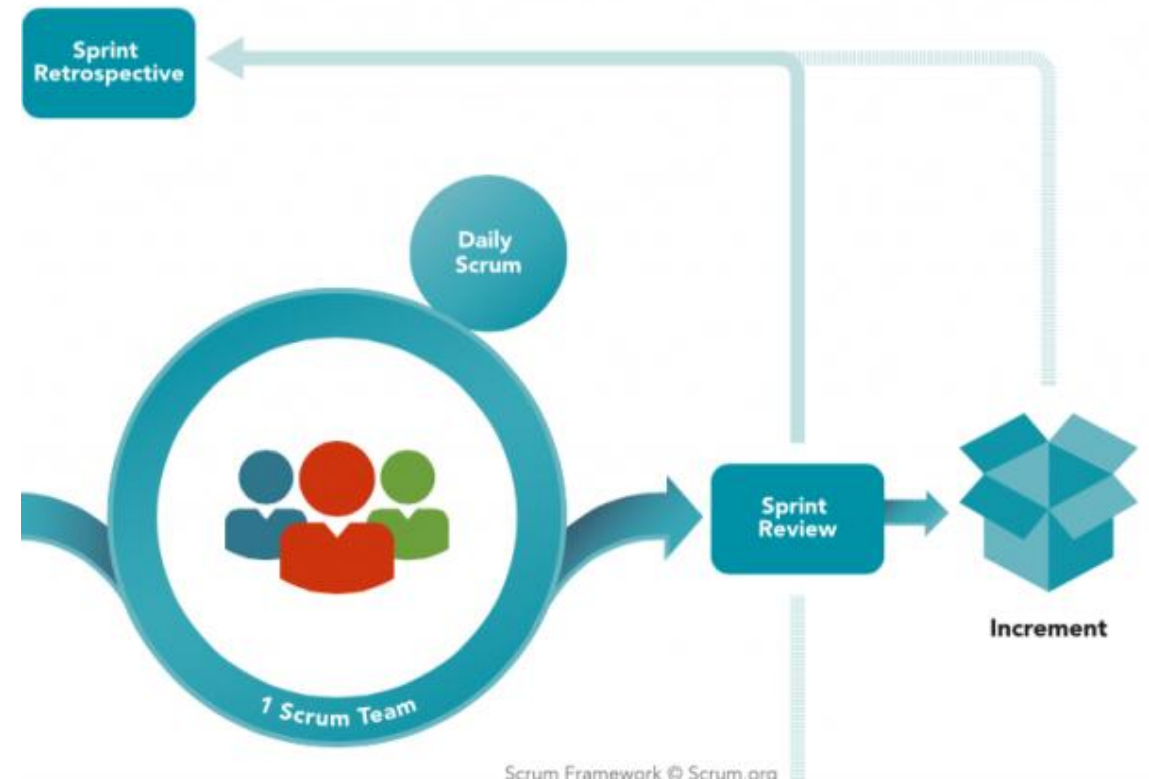


- **Historias de Usuario (HU):** Es cómo se escriben los ítems del Backlog (PB). Tienen el formato: "Como [Rol], quiero [Acción], para [Beneficio]".
- **Puntos de Historia (Story Points):** Es una métrica de esfuerzo relativo (no de tiempo). Compara la complejidad de una tarea con otra.
- **Cartas Scrum (Planning Poker):** Es la técnica para estimar los puntos. Los ingenieros votan en secreto. Evita el sesgo y fomenta el debate técnico.
- **Split (División de Historias):** Es la acción de romper una User Story grande en varias historias más pequeñas.

Desarrollo con metodologías ágiles: SCRUM

La Review

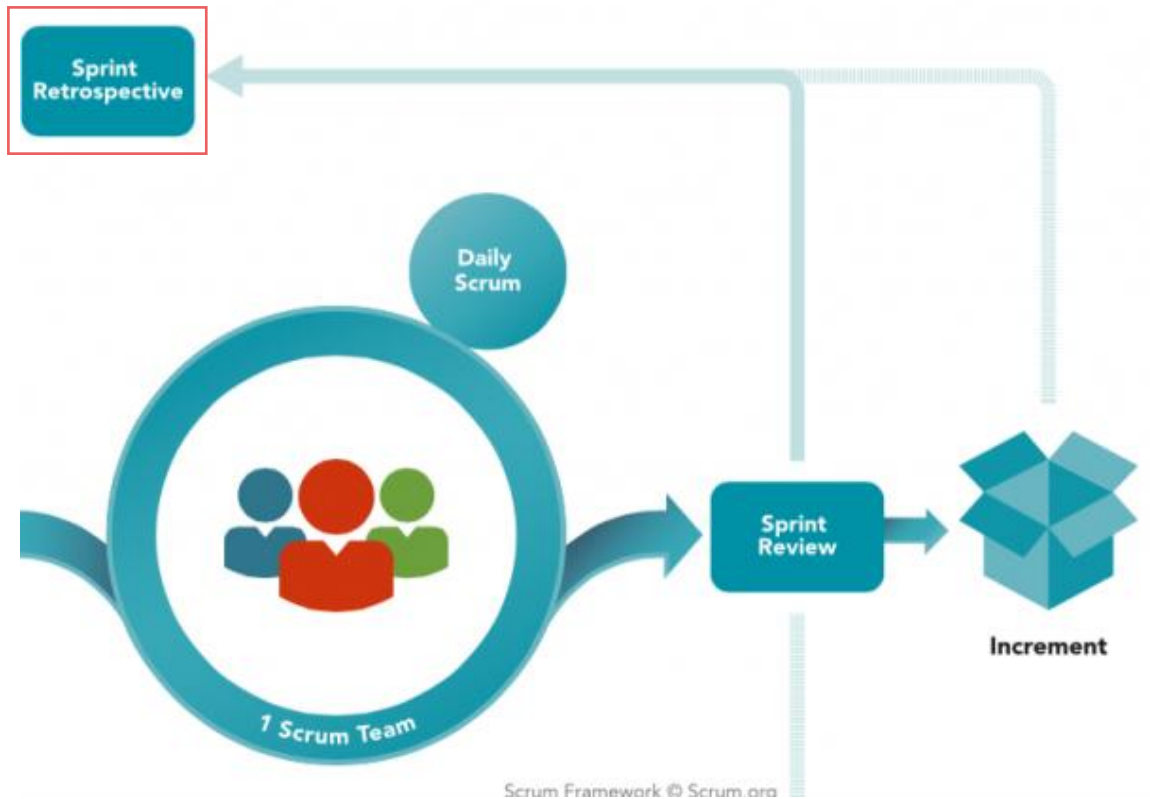
- Foco: El Incremento de producto "Hecho".
- Acción: El Scrum Team y los Stakeholders revisan el trabajo completado y el progreso hacia el Sprint Goal.
- Dinámica:
 - Muestra de software/tecnología funcionando, no transparencias.
 - El Product Owner recopila feedback y ajusta el Product Backlog.
- Resultado: Un Product Backlog actualizado para el próximo Sprint Planning.



Desarrollo con metodologías ágiles: SCRUM

La Retrospectiva

- Foco: El Scrum Team, sus interacciones, procesos y herramientas.
- Acción: Identificar éxitos, frustraciones e impedimentos técnicos u operativos.
- Dinámica (Formato Clásico):
 - ¿Qué funcionó bien? (Celebrar éxitos).
 - ¿Qué no funcionó tan bien? (Identificar problemas).
 - ¿Qué haremos diferente en el próximo Sprint? (Plan de acción).
- Resultado: Acciones de mejora concreta para implementar de inmediato.



Actividad. Planning Poker de Incertidumbre

Objetivo: aprender a dimensionar el trabajo técnico, gestionar la carga del equipo mediante visualización Kanban y entender la métrica de avance en un ciclo cerrado.

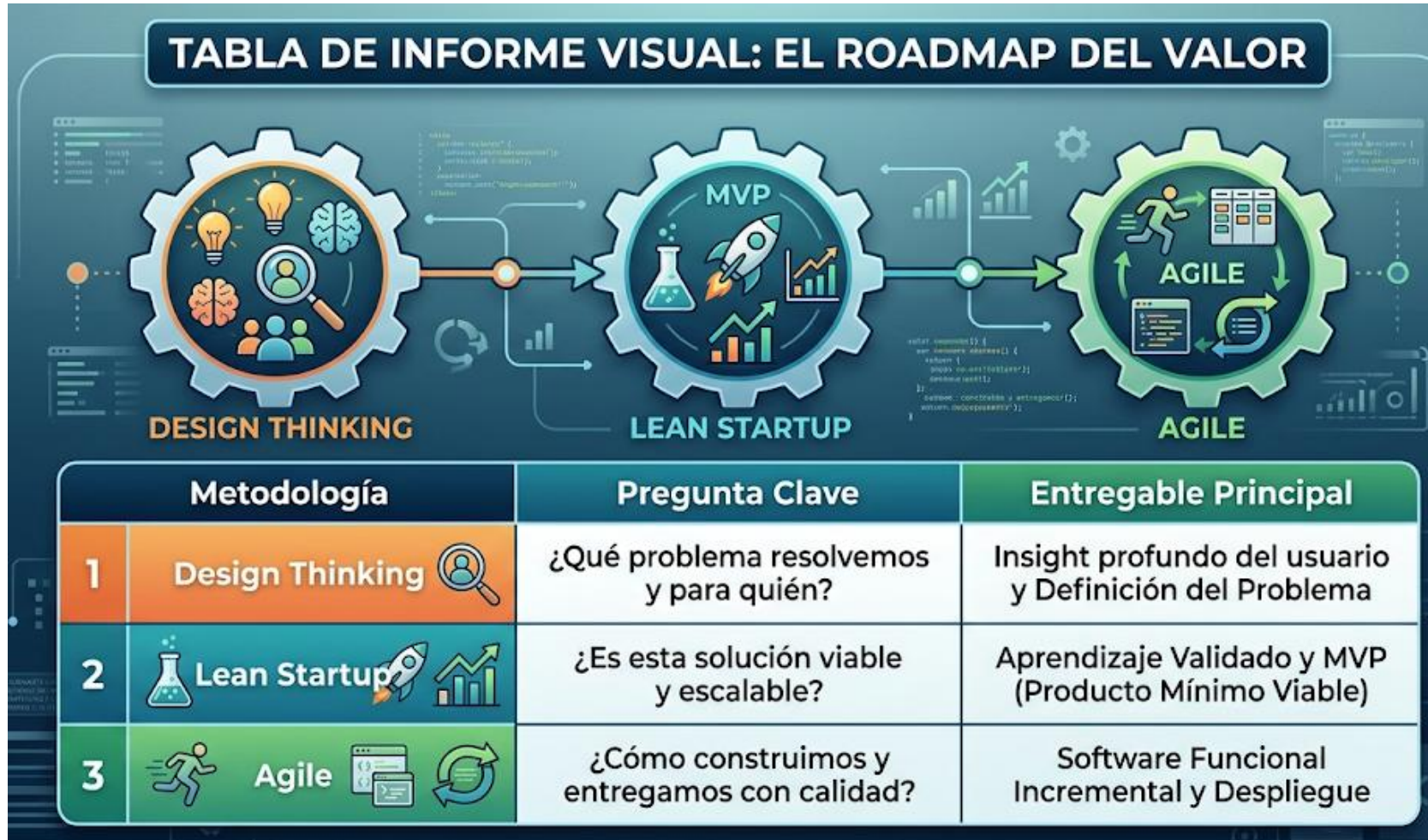


SCRUM AGILE



Desarrollo con metodologías ágiles: SCRUM

Roadmap de innovación



03

Casos de estudio y análisis de proyectos exitosos



Casos de estudio y análisis de proyectos exitosos

Unas veces se gana y otras se aprende

Este bloque muestra **cómo se han aplicado tecnologías emergentes en proyectos reales.**

Como a veces han sido un caos de éxito y como otras, a pesar de la tecnología han caído en el hype tecnológico o ha empleado metodologías que nos les han permitido validar el proyecto llevándolo al fracaso.



Casos de estudio y análisis de proyectos exitosos

Zappos (1999). ¿Cómo vender lo que no tienes?

Zappos

- **Contexto:** Año 1999. El e-commerce es incipiente. Los inversores exigen infraestructuras gigantes para confiar en un proyecto.
- **El reto técnico:** Crear un sistema de gestión de inventario, pasarela de pagos segura y logística de envíos para miles de referencias de calzado.
- **Nick Swinmurn** (fundador) no programó nada de eso. Usó una cámara de fotos.



Casos de estudio y análisis de proyectos exitosos

Dropbox (2008) – El algoritmo que no existía El vídeo de los 75.000 leads

- **Contexto:** Sincronizar archivos entre diferentes OS en 2008 era un "infierno técnico" de latencia y conflictos de archivos.
- **El reto técnico:** Resolver la concurrencia de datos en la nube para usuarios no técnicos.
- **Pregunta para el debate:** "¿Construirías primero el motor de sincronización o la interfaz de usuario? ¿Cómo validarías que la gente confiaría sus archivos a tu nube antes de escribir una sola línea de C++?"



Casos de estudio y análisis de proyectos exitosos

Segway vs. Xiaomi (Patinete) El fracaso de la ingeniería perfecta

- **Segway (\$5.000):** Estabilización por giróscopos, 5 microprocesadores, redundancia de grado militar.
- **Patinete (\$400):** Motor simple, tabla de madera/aluminio, batería de litio estándar.
- Pregunta para el debate: "Si el Segway es técnicamente superior en seguridad y equilibrio, ¿por qué terminó siendo un juguete para turistas mientras el patinete conquistó las ciudades?"



Casos de estudio y análisis de proyectos exitosos

Knight Capital (2012) – El código "Zombie" 440 millones de dólares perdidos en 45 minutos.

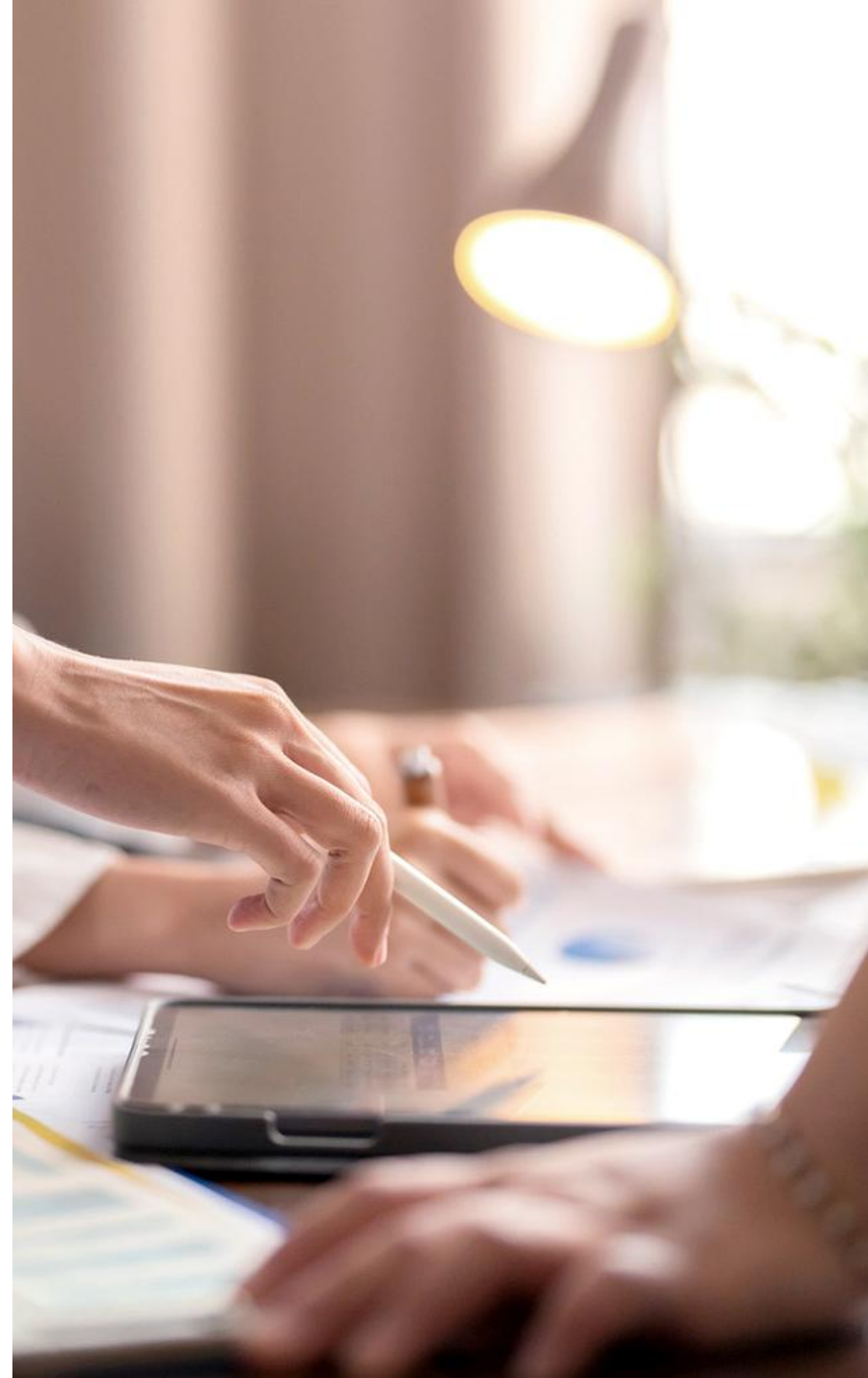
- **Escenario:** Una empresa líder en trading despliega un nuevo software de ejecución de órdenes.
- **El error:** Un servidor conservó una bandera (flag) de código antiguo de hace 8 años. El nuevo despliegue activó accidentalmente ese código "zombie".
- **Pregunta para el debate:** "¿Cómo puede un error de configuración en un solo servidor arruinar a una empresa entera? ¿Dónde falló el proceso de ingeniería?"



Casos de estudio y análisis de proyectos exitosos

SAP en LIDL (2018) – El hundimiento de los 500M€. El proyecto que nunca terminaba

- **Contexto:** LIDL intenta implantar SAP para unificar sus sistemas mundiales.
- **El Dato:** Tras 7 años de desarrollo, el sistema no era capaz de gestionar los precios de inventario de la misma forma que LIDL lo hacía históricamente.
- **Pregunta para el debate:** "¿En qué momento de esos 7 años creéis que deberían haber pivotado? ¿Por qué nadie detuvo el proyecto antes?"



Casos de estudio y análisis de proyectos exitosos

iPhone vs. BlackBerry (2007) La muerte por negación del cambio

- **Contexto:** 2007. BlackBerry tiene clientes corporativos cautivos. Apple lanza un teléfono "sin botones" y con una batería que no dura ni un día.
- **Datos comparativos:**
 - BlackBerry: Foco en la eficiencia de escritura y seguridad de red.
 - iPhone: Foco en la usabilidad y el ecosistema de aplicaciones.
- **Pregunta para el debate:** "Los ingenieros de BlackBerry se rieron del iPhone diciendo que 'nadie querría escribir en un cristal'. ¿Por qué la opinión técnica de los mejores expertos del mundo fue un error histórico?"

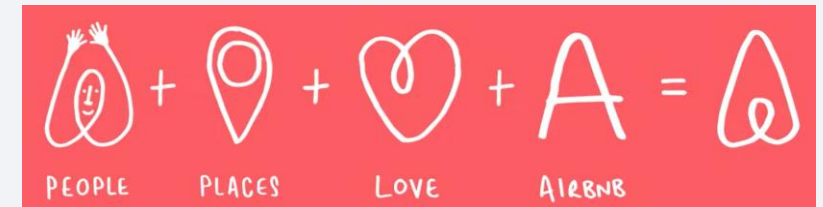


Casos de estudio y análisis de proyectos exitosos

AirBnB(2007)

El poder de la empatía visual

- **El contexto:** En 2009, la plataforma era técnicamente funcional, pero las reservas no crecían. El código era sólido, pero el producto fallaba.
- **El dato:** Tras viajar y sustituir fotos amateur por **fotografías profesionales**, los ingresos por semana pasaron de 200\$ a 400\$ en pocos días.
- **Pregunta para el debate:** Si los datos dicen que el producto falla, ¿cuántas veces buscamos la solución en el *refactoring* del código en lugar de en la experiencia del usuario?



Casos de estudio y análisis de proyectos exitosos

Facebook (2004)

El "Hacker Way" del MVP

- **El contexto:** Un equipo de ingenieros dedicó años a un motor de videojuegos (*Glitch*) que nadie quería jugar, pero construyeron un chat interno para coordinarse.
- **El dato:** Decidieron cerrar el juego y **descartar el 90% de su base de código** para centrarse solo en la herramienta de chat.
- **Pregunta para el debate:** ¿Estamos preparados psicológicamente para "tirar a la basura" meses de trabajo técnico si descubrimos que el valor está en una funcionalidad secundaria?



Casos de estudio y análisis de proyectos exitosos

Slack

La maestría del pivote técnico

- **Problema:** El producto principal (videojuego *Glitch*) no era rentable.
- **Acción:** Identificación de una herramienta interna valiosa + Eliminación del código legacy.
- **Clave:** Flexibilidad mental para "matar" tu proyecto si el valor está en otro sitio.
- **Resultado:** De videojuego fallido a estándar de comunicación corporativa.



Casos de estudio y análisis de proyectos exitosos

Toyota

El origen de la mejora continua (Kaizen)

- **El contexto:** En producción masiva, un error pequeño se multiplica por miles. La mayoría de las empresas ocultan fallos para no parar la cadena.
- **El dato:** El método de los **5 Porqués** redujo sus defectos de fabricación a niveles históricos, convirtiéndolos en la empresa de motor más eficiente del mundo.
- **Pregunta para el debate:** En desarrollo de software, ¿hacemos autopsias de errores para buscar culpables o para encontrar fallos en el proceso?



TOYOTA

Casos de estudio y análisis de proyectos exitosos

Google Glass Ingeniería sin propósito

- **El contexto:** Un despliegue de ingeniería óptica y miniaturización sin precedentes. Un producto "perfecto" desde el punto de vista del hardware.
- **El dato:** Se lanzó sin una "**Killer App**" (una utilidad real diaria), lo que provocó que el público las viera como un juguete caro e invasivo.
- **Pregunta para el debate:** ¿Puede la excelencia técnica compensar la falta de un problema real que resolver?



Casos de estudio y análisis de proyectos exitosos

Fracaso por Exceso de Ingeniería: El caso Quibi



- **El contexto:** Construyeron una plataforma de streaming con tecnología propietaria para rotar video sin latencia (Turnstyle) y contenido de Hollywood.
- **El dato:** Gastaron **1.750 millones de dólares** antes de tener el primer usuario activo. Cerraron en 6 meses.
- **Pregunta para el debate:** Como ingenieros, ¿cómo podemos evitar que el ego tecnológico nos lleve a construir sistemas hiperscalables para un mercado que no existe?

Casos de estudio y análisis de proyectos exitosos

KODAK

La muerte por negación del cambio

- **El contexto:** Un ingeniero de Kodak inventó la primera cámara digital en 1975. La empresa tenía el conocimiento y la infraestructura.
- **El dato:** La directiva bloqueó la tecnología para **proteger el margen de beneficio** de los carretes de película.
- **Pregunta para el debate:** Si nuestra tecnología actual "canibaliza" el producto estrella de nuestra empresa, ¿debemos liderar el cambio o esperar a que lo haga la competencia?



Casos de estudio y análisis de proyectos exitosos

IKEA

La muerte por negación del cambio

- **El contexto:** La AR solía verse como un truco de marketing (filtros de cara). IKEA la aplicó a la logística de mobiliario.
- **El dato:** Los modelos 3D tienen un **98% de precisión de escala**, permitiendo al usuario "ver" el mueble en su casa antes de comprarlo.
- **Pregunta para el debate:** ¿Es mejor usar una tecnología emergente como reclamo publicitario o como herramienta para reducir costes operativos (devoluciones)?



¡Gracias!