

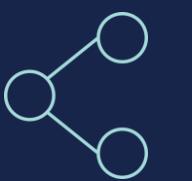
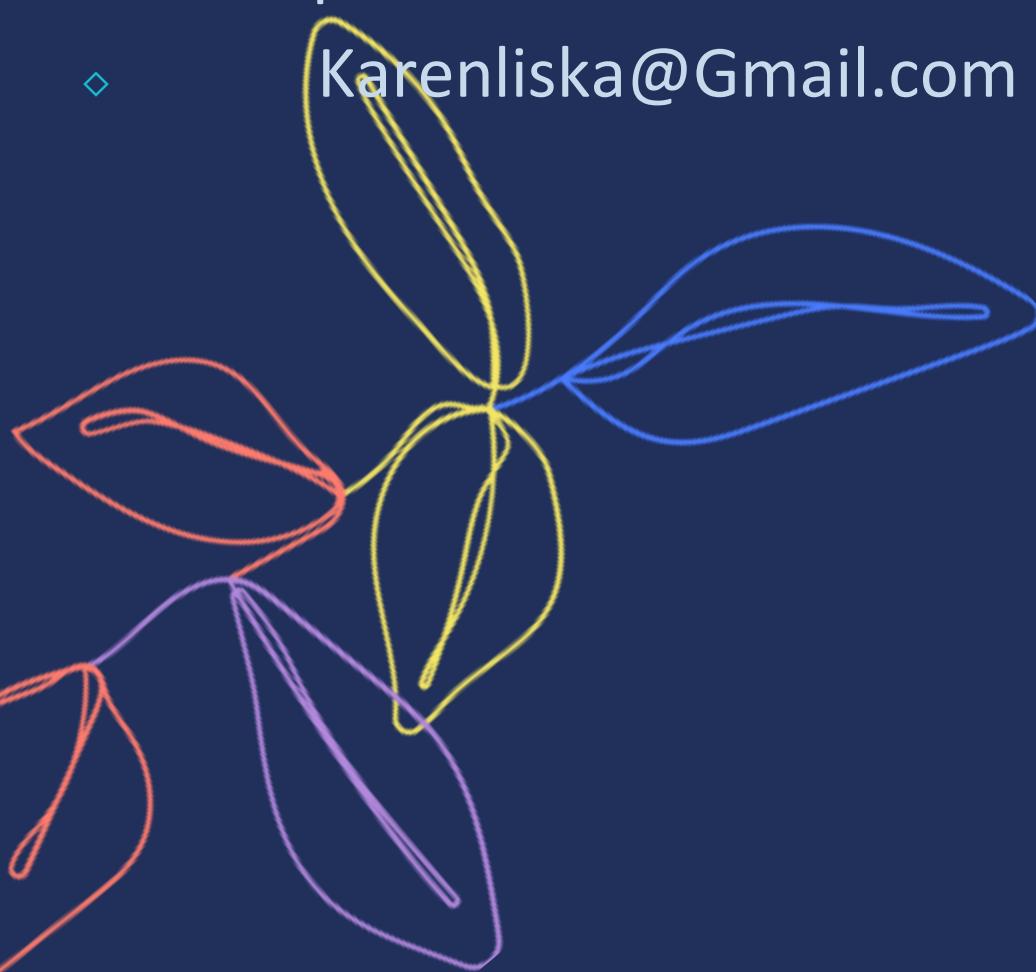
Ingeniería de Software 1



Hola!

Karen Liska:

- Ing. En Informática y Sistemas
- Mgtr. Seguridad Informática
- Experiencia en desarrollo de software
- Experiencia en calidad de software
- Karenliska@gmail.com



01 Objetivos



Objetivos

Objetivo 1

Identificar los aspectos que vuelven al software una disciplina de la ingeniería.

Objetivo 2

Reconocer la importancia del control de calidad junto a sus diferentes técnicas como parte de esta rama de la ingeniería

Objetivo 3

Distinguir las diferentes etapas del ciclo de vida de software y su correcta aplicación.

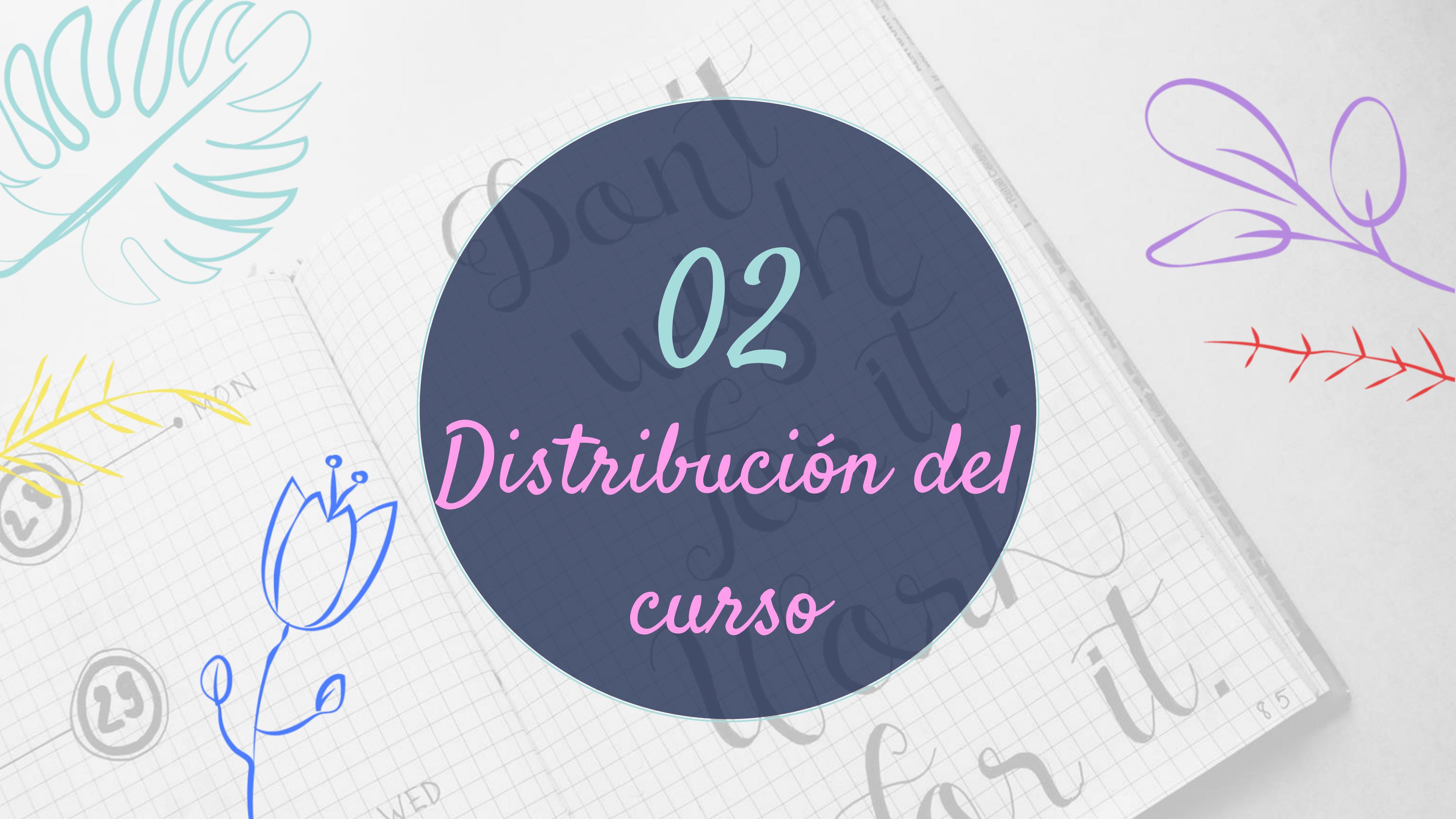
Objetivo 4

Poner en práctica la administración de recursos en un proyecto de software.

02

Distribución del

curso



Objetivos

Evaluación Parcial

30

Evaluaciones Cortas

15

Trabajo en Clase

10

Laboratorio

15

Examen Final

30



Manos a la obra



Organizar

Plan para proyecto de software



Experimentar

Utilizar herramientas del mercado



Proponer

Evaluar diferentes formas de
crear un producto de software

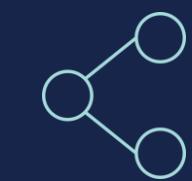
03

Algunos conceptos

Ing. De SW vs Ing en Informática y sistemas



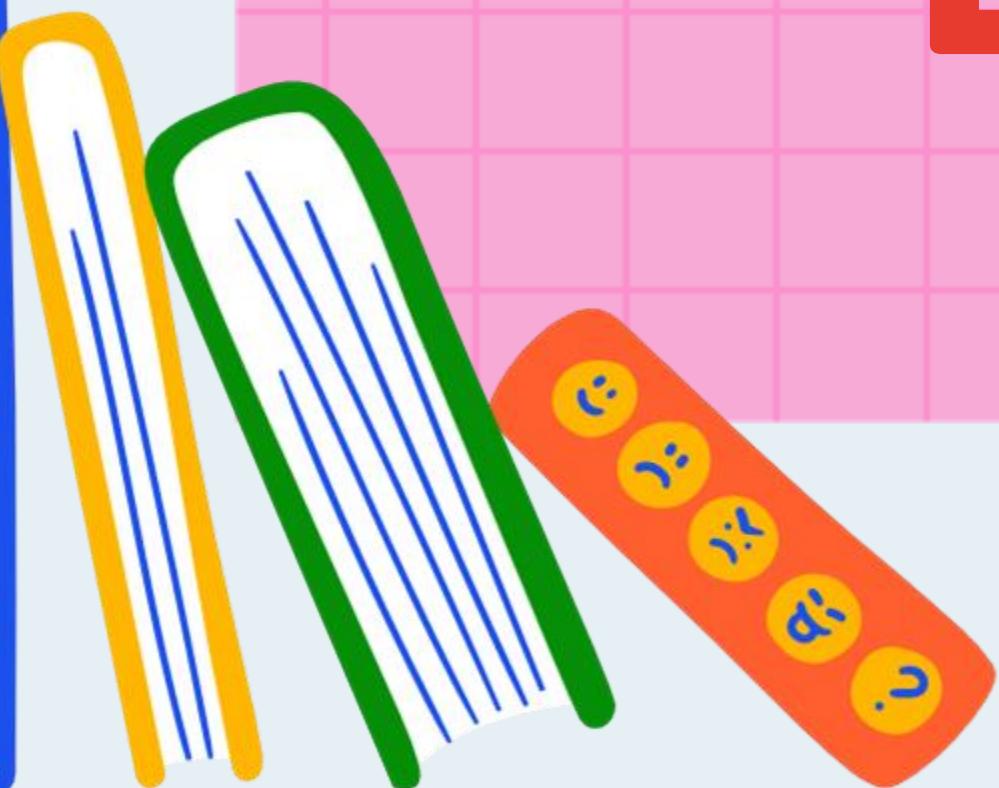
Programadores, desarrolladores, ingenieros



Unidad 1

El Software en los negocios

MEMORIAS



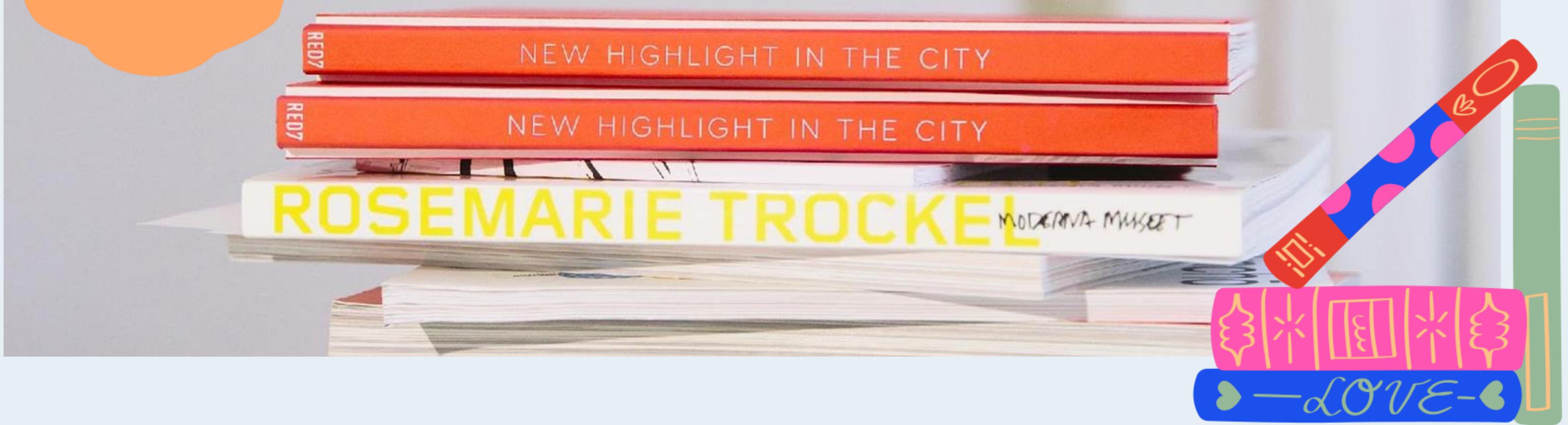
e



Repasando Conceptos

Desarrollador ...

Programador ...

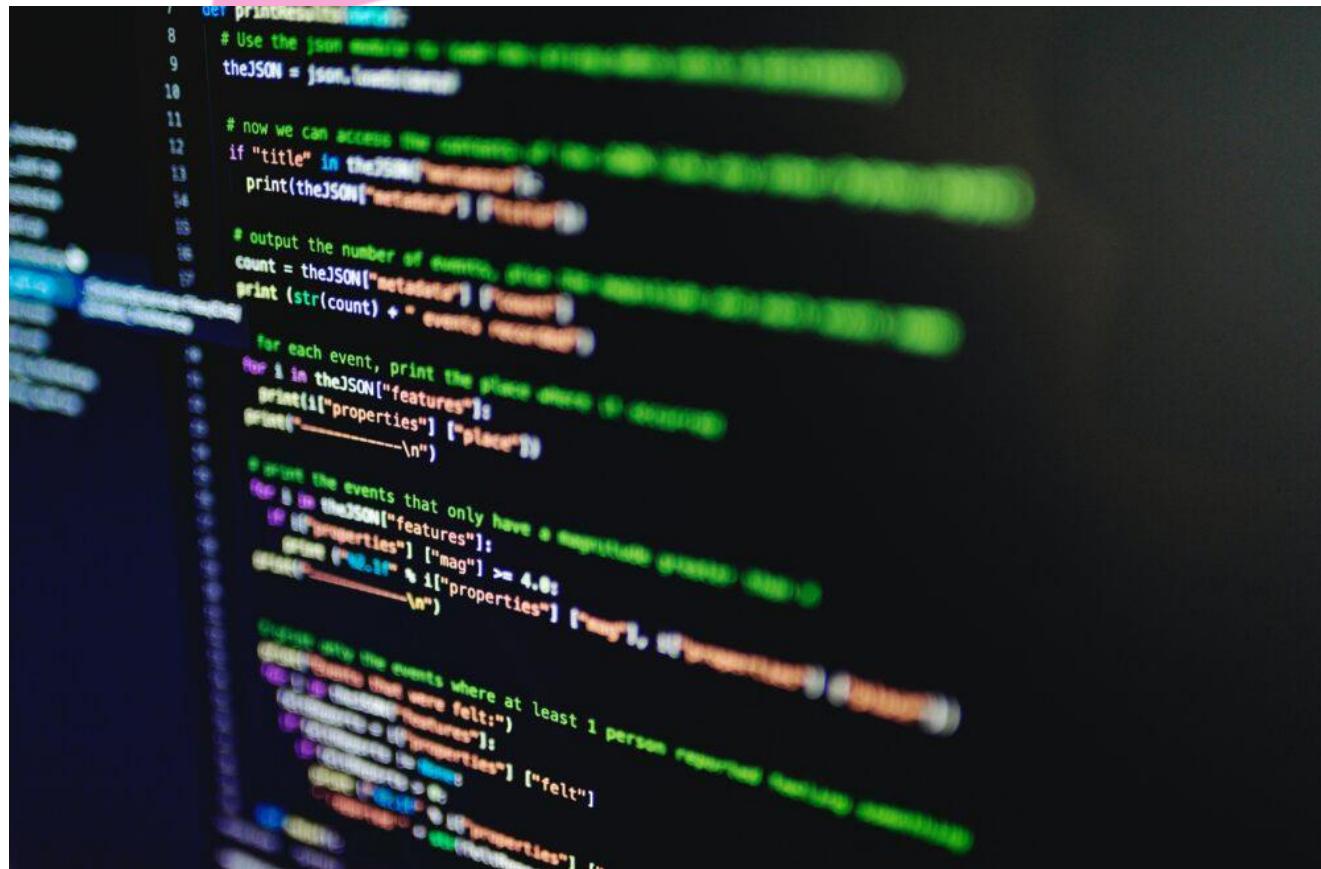




**Todo Ingeniero de SW es un
Programador mas no todo
Programador es un Ingeniero
de Software.**

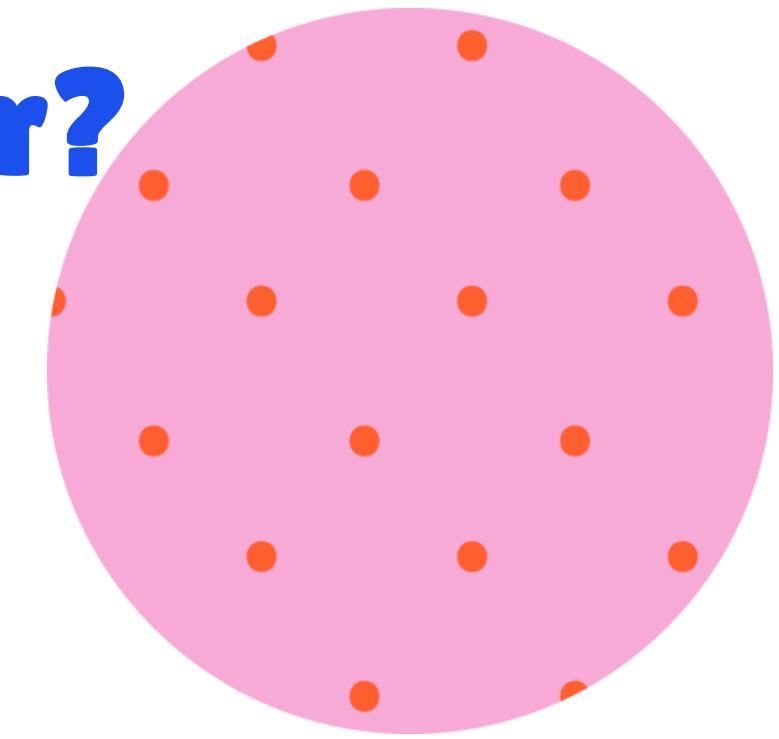
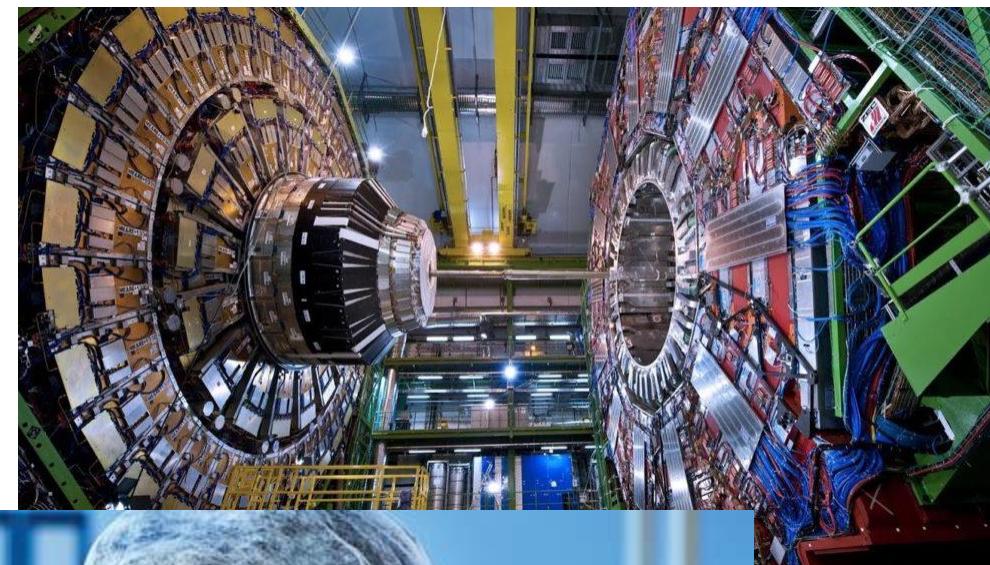
Programador

- Profundiza en uno o más lenguajes de programación.
- Librerías
- Conoce qué se puede y no hacer con los lenguajes o tecnologías que domina.
- Hace sus propias pruebas unitarias
- Ve los requerimientos acompañados de su respectivo diseño.



```
7 def printMetadata():
8     # Use the json module to read the file
9     theJSON = json.loads(data)
10
11     # now we can access the contents of the JSON object
12     if "title" in theJSON["metadata"]:
13         print(theJSON["metadata"]["title"])
14
15     # output the number of events, after removing missing data
16     count = theJSON["metadata"]["count"]
17     print(str(count) + " events received")
18
19     # for each event, print the place where it occurred
20     for i in theJSON["features"]:
21         print(i["properties"]["place"])
22         print("\n")
23
24     # print the events that only have a magnitude greater than or equal to 4.0
25     for i in theJSON["features"]:
26         if i["properties"]["mag"] >= 4.0:
27             print(i["properties"]["mag"], i["place"])
28             print("\n")
29
30     # print out the events where at least 1 person reported feeling them
31     for i in theJSON["features"]:
32         if i["properties"]["felt"] >= 1:
33             print(i["properties"]["mag"], i["place"], i["properties"]["felt"])
34             print("\n")
```

Ouién duede ser un programador?



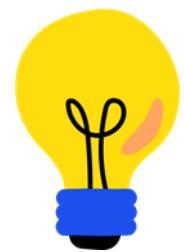
Code Written By A Math Ph.D.

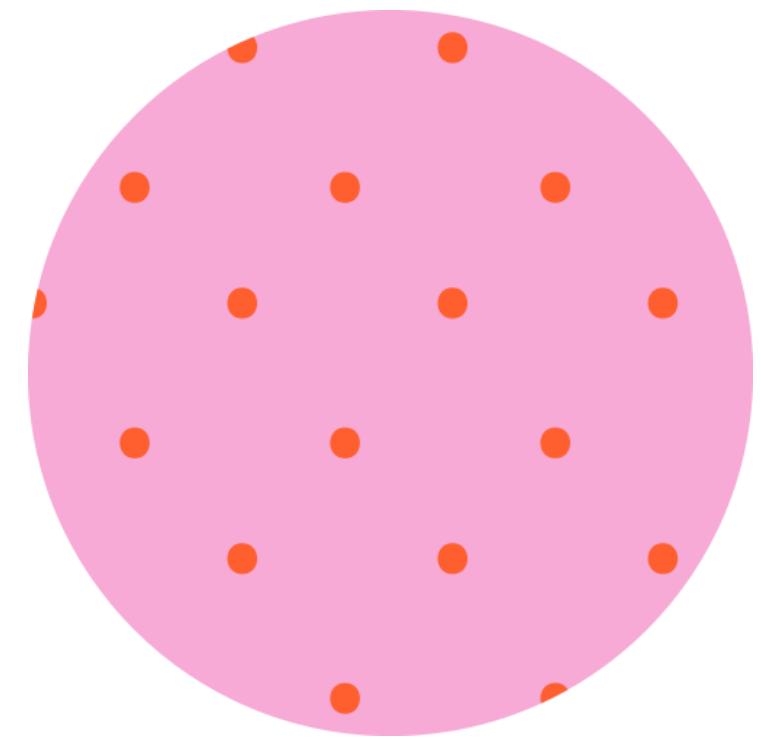
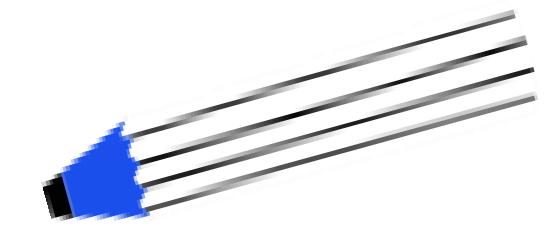
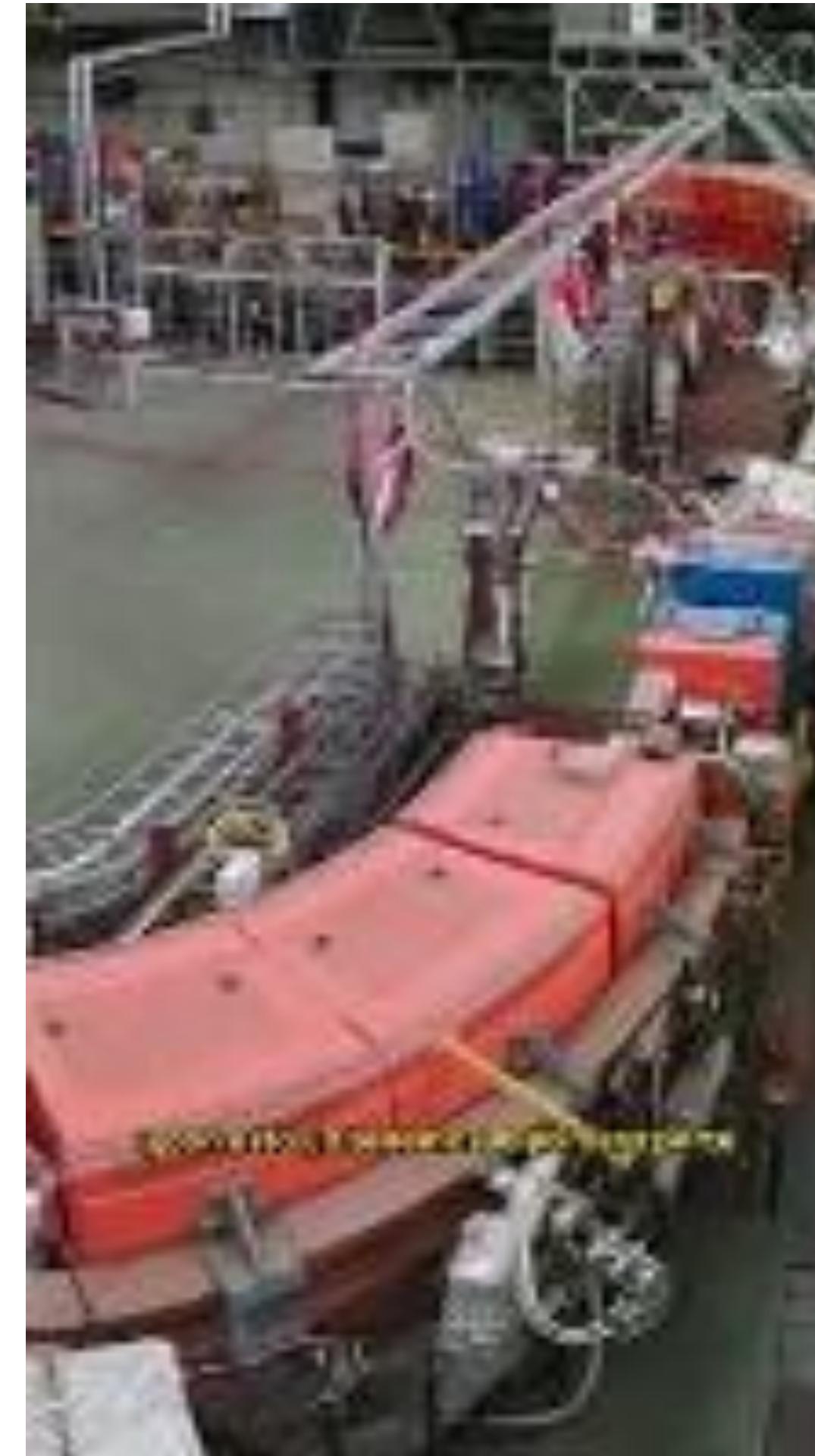
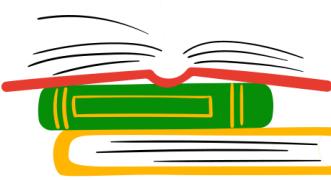
```
public int getFibonacciNumber(int n) {
    return (int) divide(subtract(exponentiate(phi(), n), exponentiate(psi(), n)),
        subtract(phi(), psi()));
}

public double exponentiate(double a, double b) {
    if (equal(b, zero())) {
        return one();
    } else {
        return multiply(a, exponentiate(a, subtract(b, one())));
    }
}

public double phi() {
    return divide(add(one(), sqrt(add(one(), one(), one(), one(), one()))),
        add(one(), one()));
}

public double psi() {
    return subtract(one(), phi());
}
```





Conocimientos de un Ing. de SW.



- Programación
- Sistemas Operativos
- Redes
- Bases de datos
- IA
- Diseño de algoritmos
- Planificación
- Seguridad
- Planificación de control de calidad
- Patrones de diseño
- Integración entre sistemas
- Metodologías de desarrollo
- etc.



Ciencias de la computación

Por qué es importante?



- Diagnóstico de problemas
- Evalúa requerimientos basado en el impacto en el proyecto
- Solución de problemas de forma eficiente
- Software flexible y fácil de mantener
- Relación con el negocio
- Habilidades de planificación
- Habilidades de formación de equipos
- Resolución de problemas avanzada
- Gestión de proyectos
- Etc.

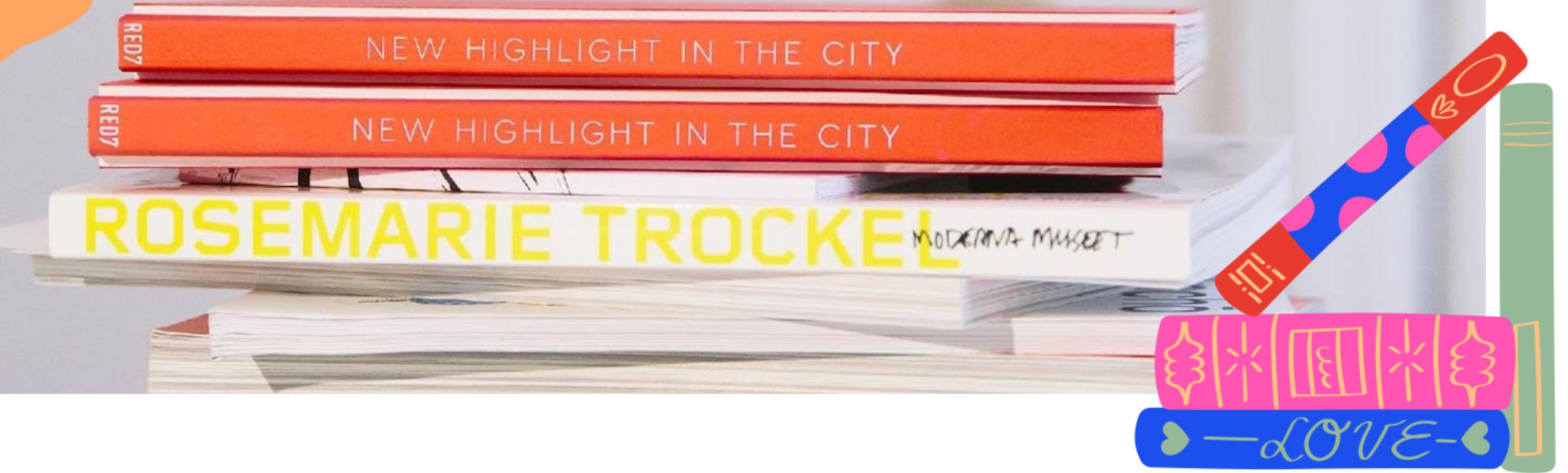


**Un Ingeniero de Software
participa en cada etapa del
ciclo de vida del software.**

Repasemos un
poco...

Análisis...

Diseño... Ejecución



Pasos generales del ciclo de vida

1

Análisis del negocio

2

**Análisis de
requerimientos**

3

Diseño

4

Desarrollo

5

Pruebas

6

**Implementación y
monitoreo**



Puestos de un Ingeniero de Software



- Product Owner
- Project Manager
- Business Analyst
- Desarrollador
- Arquitecto de Software
- Director de proyectos de software
- Líder de equipos de software
- Arquitecto de Control de calidad
- Auditor
- Business Intelligence

Existen 2 formas de crear software.

Software que funciona

No escalable

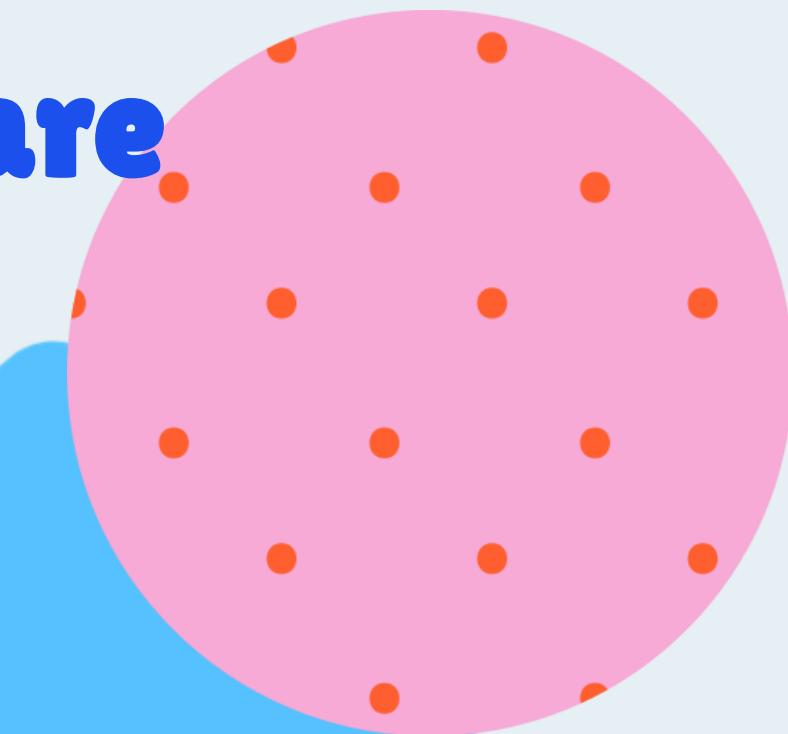
Hecho pensando en un entregable

Software con alta calidad

Seguro

Adaptable al cambio

Documentado



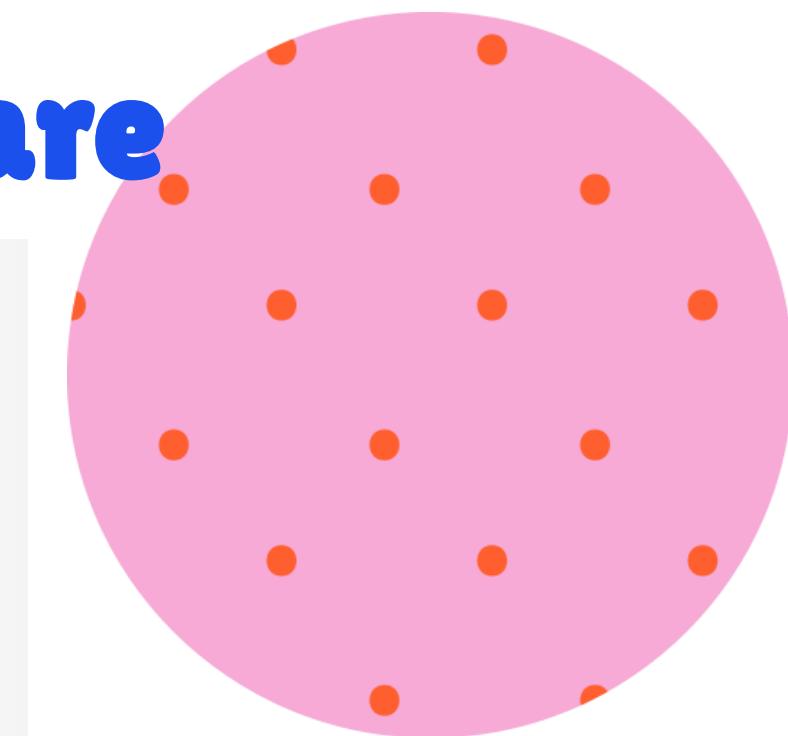
Existen 2 formas de crear software.

Code Written By A CS 101 Student

```
public int fibonacci(int x) {  
    if (x == 1) {  
        return 1;  
    } else if (x == 2) {  
        return 1;  
    } else {  
        return fibonacci(x - 1) + fibonacci(x - 2);  
    }  
}
```

Code Written At A Hackathon

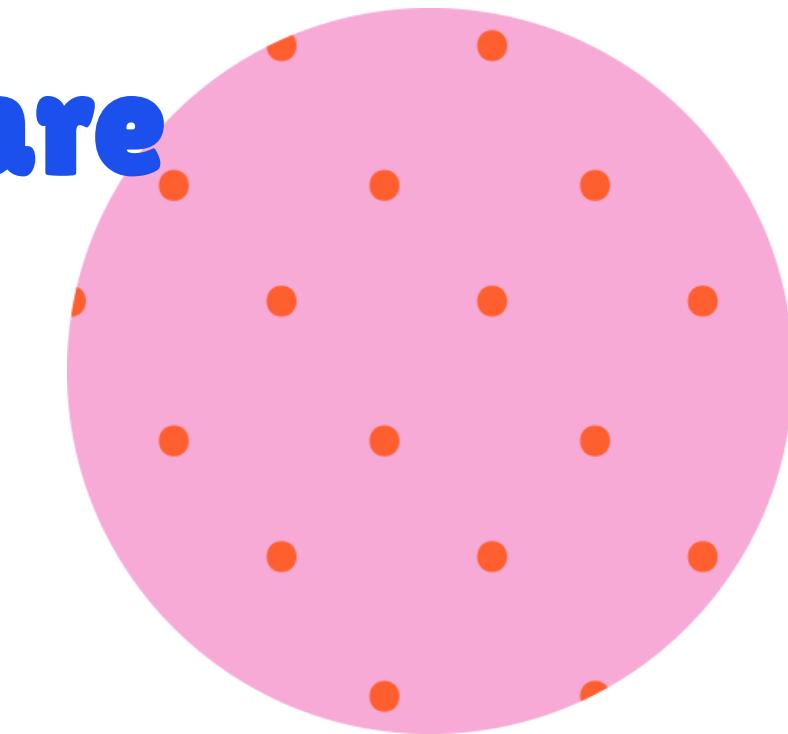
```
public int getFibonacciNumber(int n) {  
    switch(n) {  
        case 1: return 1;  
        case 2: return 1;  
        case 3: return 2;  
        case 4: return 3;  
        case 5: return 5;  
        case 6: return 8;  
        case 7: return 13;  
    default:  
        // good enough for the demo, lol  
        return -1;  
    }  
}
```



Existen 2 formas de crear software.

Code Written At A Startup

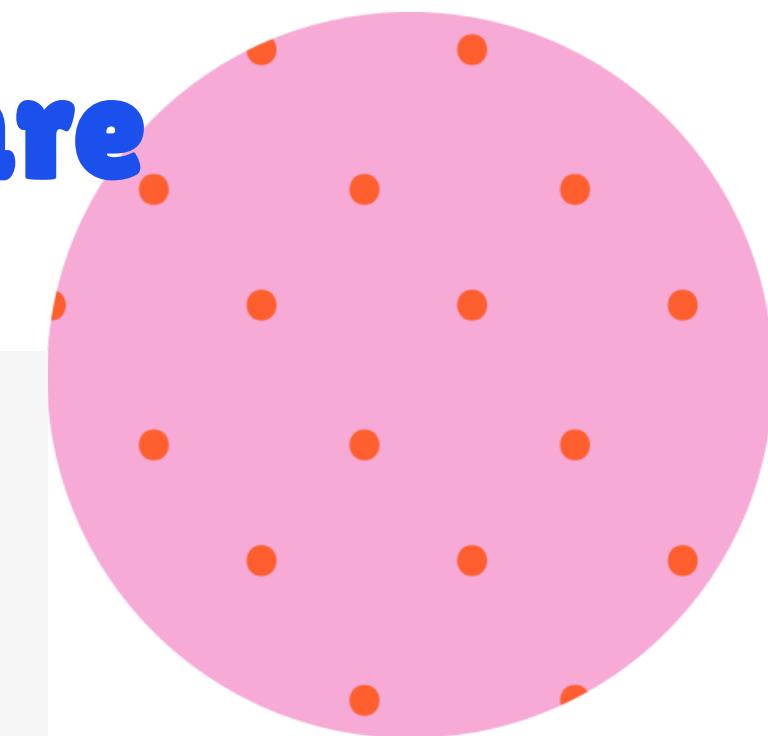
```
// TODO add Javadoc comments
/**
 * getFibonacciNumber
 */
// TODO Should we move this to a different file?
public int getFibonacciNumber(int n) {
    // TODO Stack may overflow with recursive implementation, switch over to
    // iteration approach at some point?
    if (n < 0) {
        // TODO This should probably throw an exception. Or maybe just print
        // a log message?
        return -1;
    } else if (n == 0) {
        // TODO Generalize the initial conditions?
        return 0;
    } else if (n == 1) {
        return 1;
    } else {
        // TODO Spend some time with my family and kids, I've been at work for
        // over 48 hours straight.
        return getFibonacciNumber(n - 1) + getFibonacciNumber(n - 2);
    }
}
```



Existen 2 formas de crear software.

Code Written At A Large Company

```
/**
 * getFibonacciNumber is a method that, given some index n, returns the nth
 * Fibonacci number.
 * @param n The index of the Fibonacci number you wish to retrieve.
 * @return The nth Fibonacci number.
 */
public CustomInteger64 getFibonacciNumber(CustomInteger64 n) {
    FibonacciDataViewBuilder builder =
        FibonacciDataViewBuilderFactory.createFibonacciDataViewBuilder(
            new FibonacciDataViewBuilderParams(n, null, null, 0, null));
    if (builder == FibonacciDataViewBuilderConstants.ERROR_STATE) {
        throw new FibonacciDataViewBuilderFactoryException();
    }
    FibonacciDataView dataView = builder.GenerateFibonacciDataView(this);
    if (dataView == FibonacciDataViewConstants.ERROR_STATE) {
        throw new FibonacciDataViewGenerationException();
    }
    return dataView.accessNextFibonacciNumber(null, null, null);
}
```





**El resultado de un programador
no es necesariamente un
producto terminado**

Qué hace que el
SW sea una
Ingeniería?



Software como Ingeniería

Disciplina	Implica el uso de estructuras, herramientas y técnicas para construir productos de software
Análisis	Analiza diferentes situaciones para brindar una solución con el software
Es sistemática	Tiene un ciclo de vida y diferentes metodologías que se adaptan a este.
Resuelve otros problemas	Resuelve problemas de costos, tiempos entre otras necesidades.
Requiere ingenio	No solo aplica una tecnología, la crea de ser necesario.



Ingenieria

Herramientas de la ingeniería

Procesos

Fórmulas

Estrategias

Marco de trabajo

Construir!
Ingenio!



Ingenieria de software

Herramientas de la ingeniería

Procesos

Metodologías/Patrones de diseño –

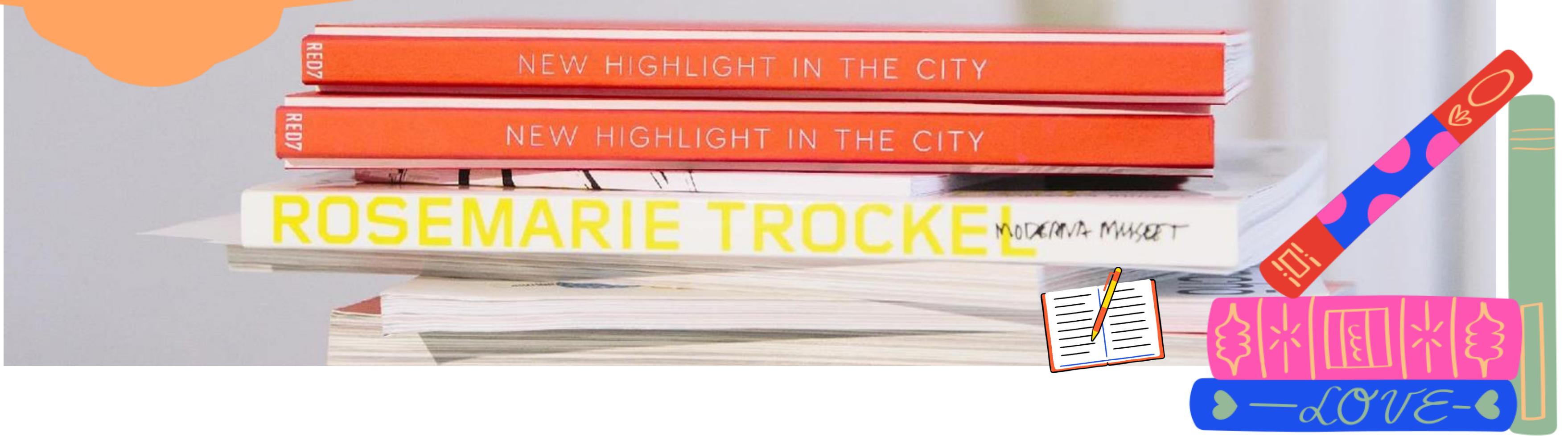
(Fórmulas/Estrategias)

Marco de trabajo

Construir!
Ingenio!



El Problema del Software



Software como giro del negocio

- Existen empresas que se dedican por completo a fabricar software.
- Venden software a medida o de ciertos tipos para diferentes clientes.
- Se han especializado en un tipo de software o sistema de gestión y lo venden.

Otros giros de negocio

- Existen empresas que su función no es principalmente fabricar Software.
- En el mundo de hoy, todas las empresas necesitan software en cierta medida para funcionar.
- La información es tanta, que es imposible manejarla sin software.

Pero...

- El Software representa un gasto
- Muchas veces es difícil que se vean los beneficios del software a corto plazo
- Existe Resistencia al cambio en muchas empresas.

Ingeniero de Software

Analiza procesos



Siguiendo los pasos de
la Ingeniería

Crea soluciones



Para mejorar los
procesos, como otras
ingenierías.

**Propone
soluciones**

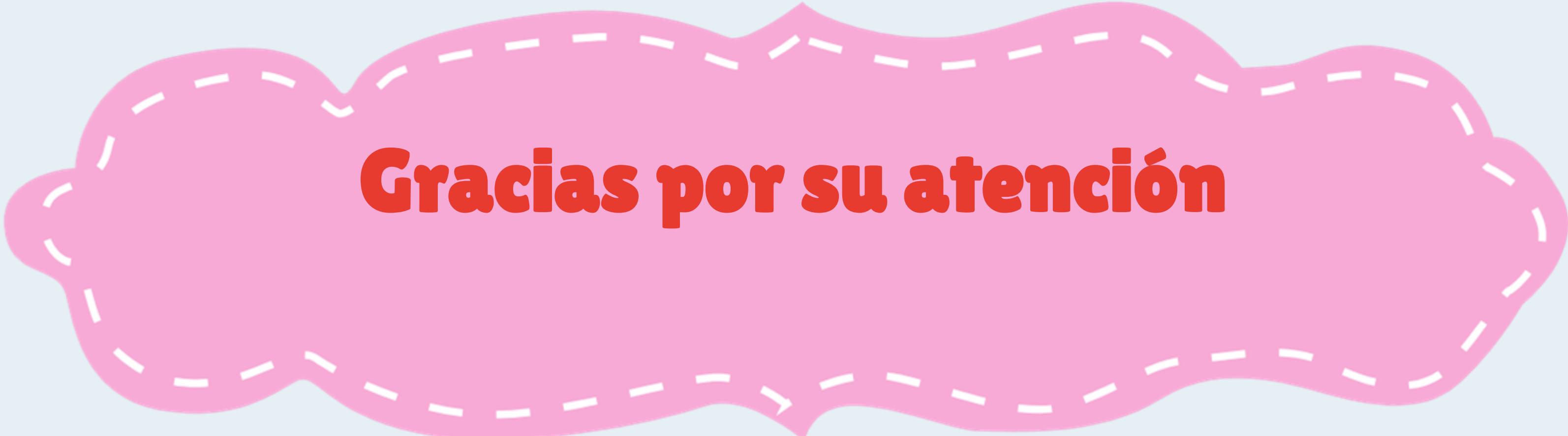


Negocia con el cliente

**Implementa la
solución**



Ejecuta y Mantiene.



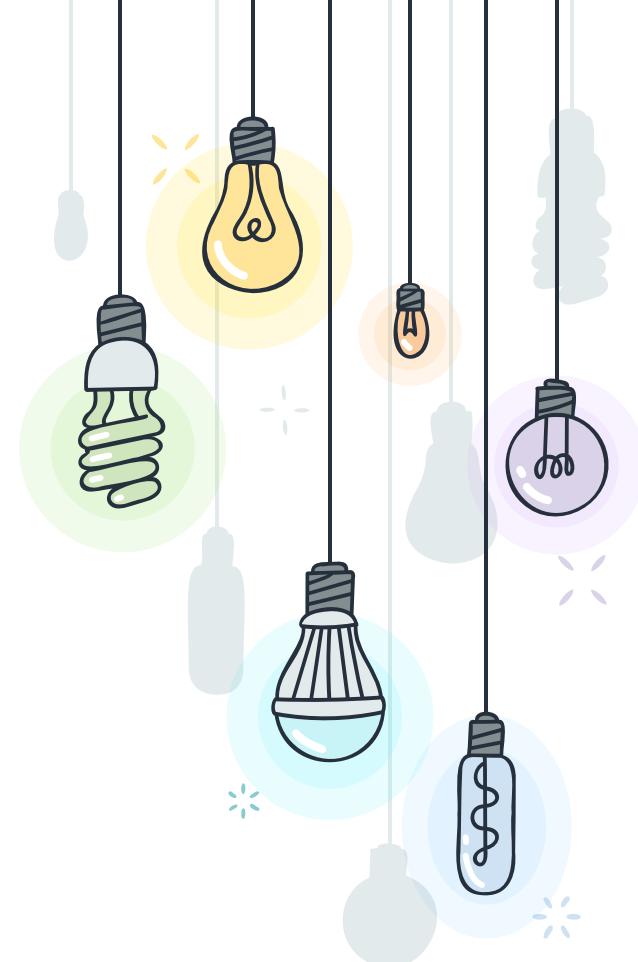
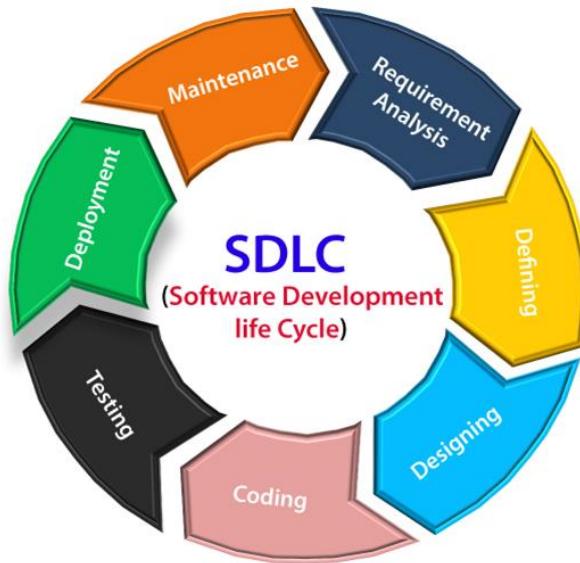
Gracias por su atención



SDLC

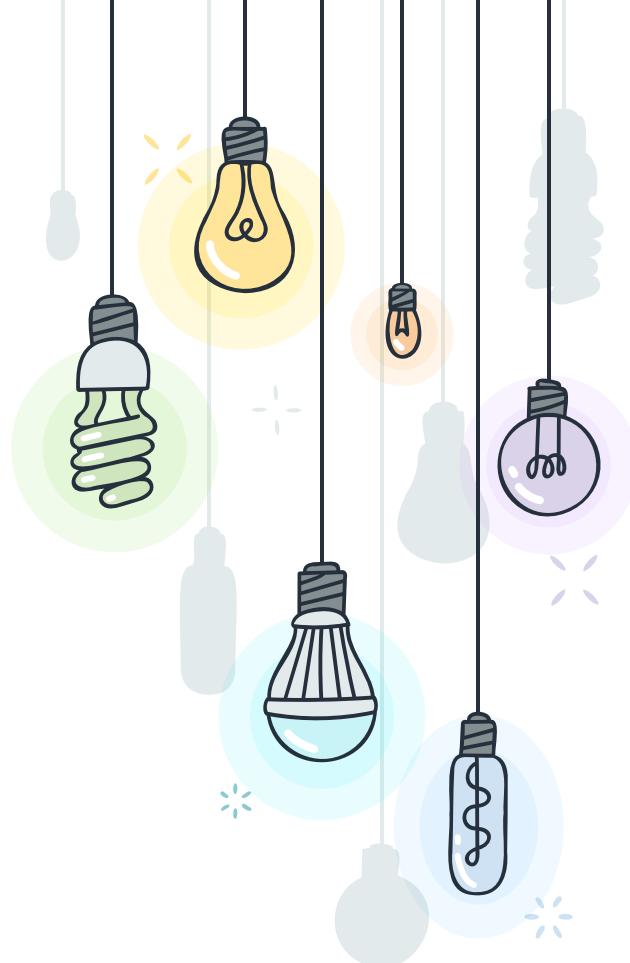
* EN CONCRETO ¿QUÉ ES?

- + Es un vistazo global de todos los pasos involucrados en la creación de software.



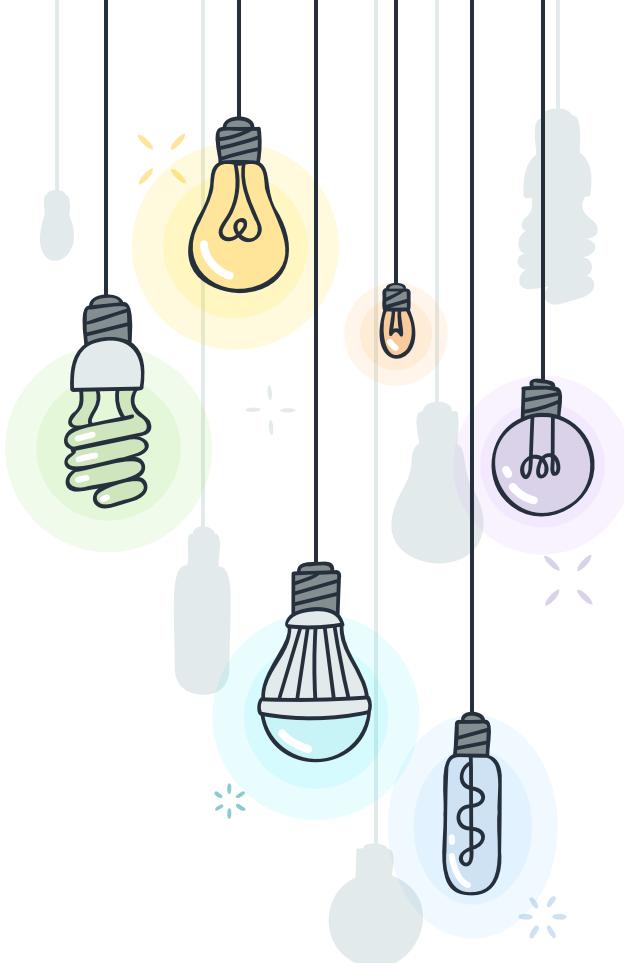
* EN CONCRETO ¿QUÉ ES?

- + Es un proceso
- + Es una estrategia
- + Es una fórmula
- + Es un marco de trabajo
- + **No es una metodología**



* SUS OBJETIVOS

- + Reducir el riesgo del proyecto
- + Cumplir con las expectativas del cliente
- + Que el software sea de alta calidad tanto durante su Desarrollo como su utilización
- + Costo efectivo
- + Tiempo eficiente



LAS METODOLOGÍAS
DE DESARROLLO
SURGEN DE ESTO



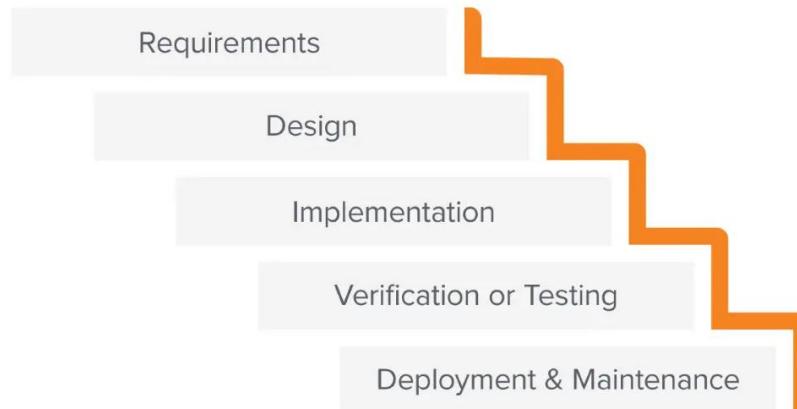
1

REPASANDO UN POCO...



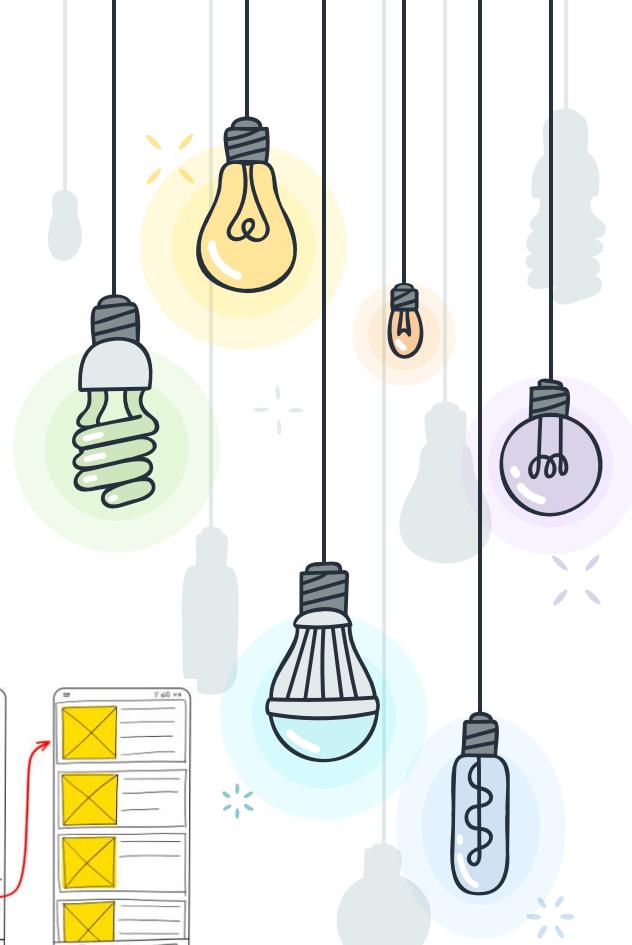
CASCADA

- + Análisis de requerimientos
- + Diseño
- + Desarrollo
- + Pruebas
- + Implementación
- + Mantenimiento



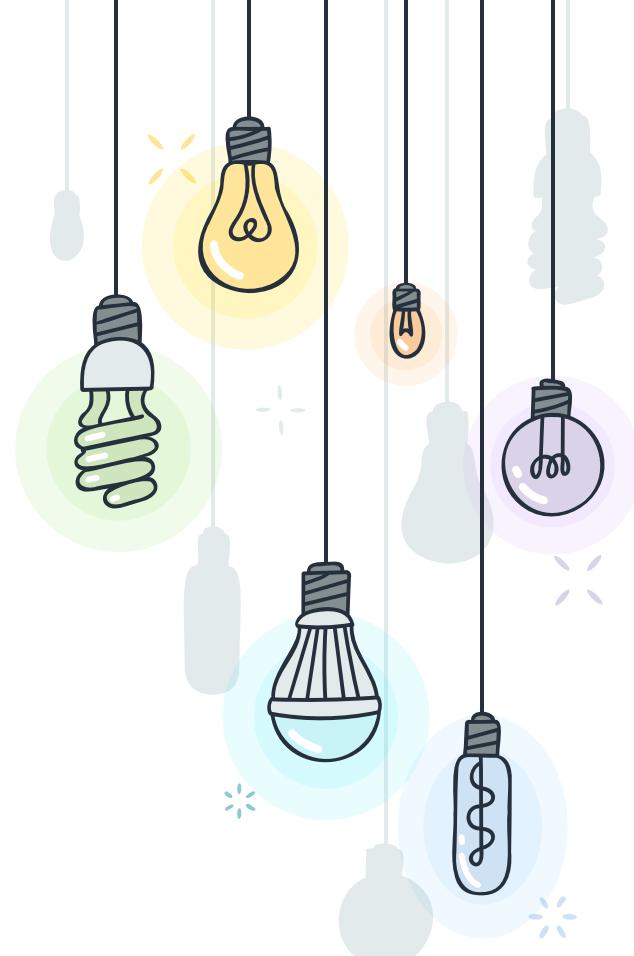
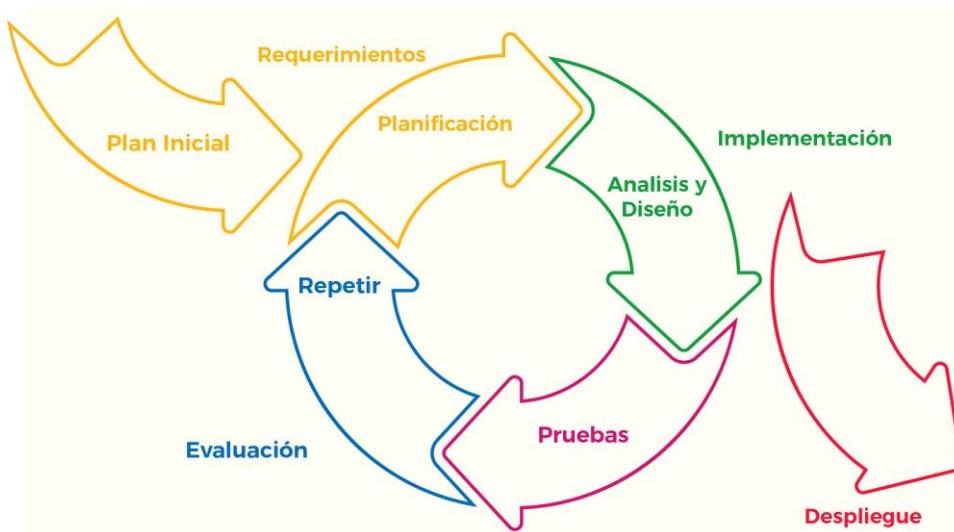
✳️ PROTOTIPOS

- + Análisis de requerimientos
- + Diseño
- + Desarrollo de prototipo
- + Evaluación del cliente
- + Refinar el prototipo
- + Desarrollo
- + Pruebas...

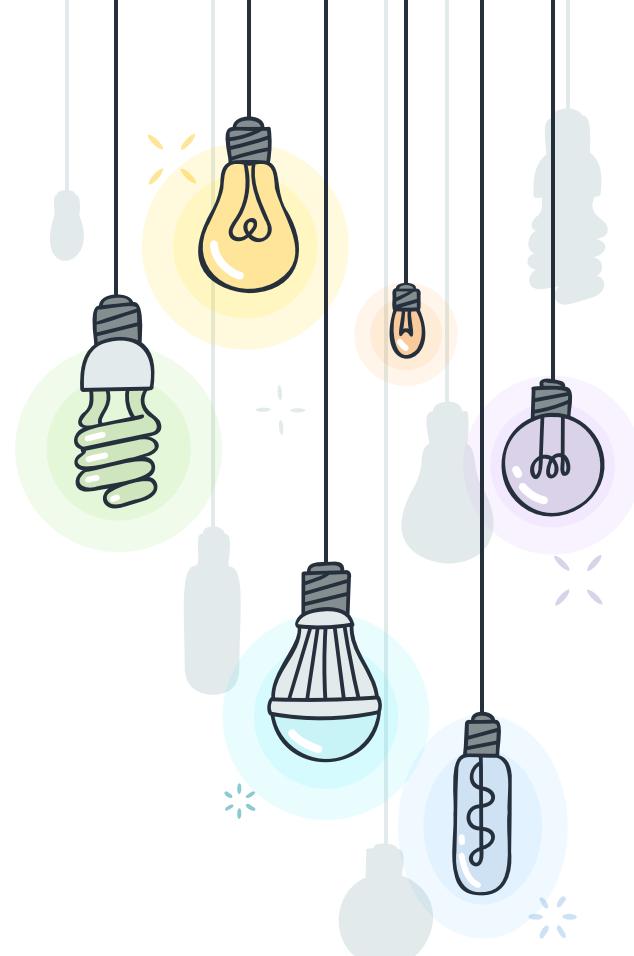
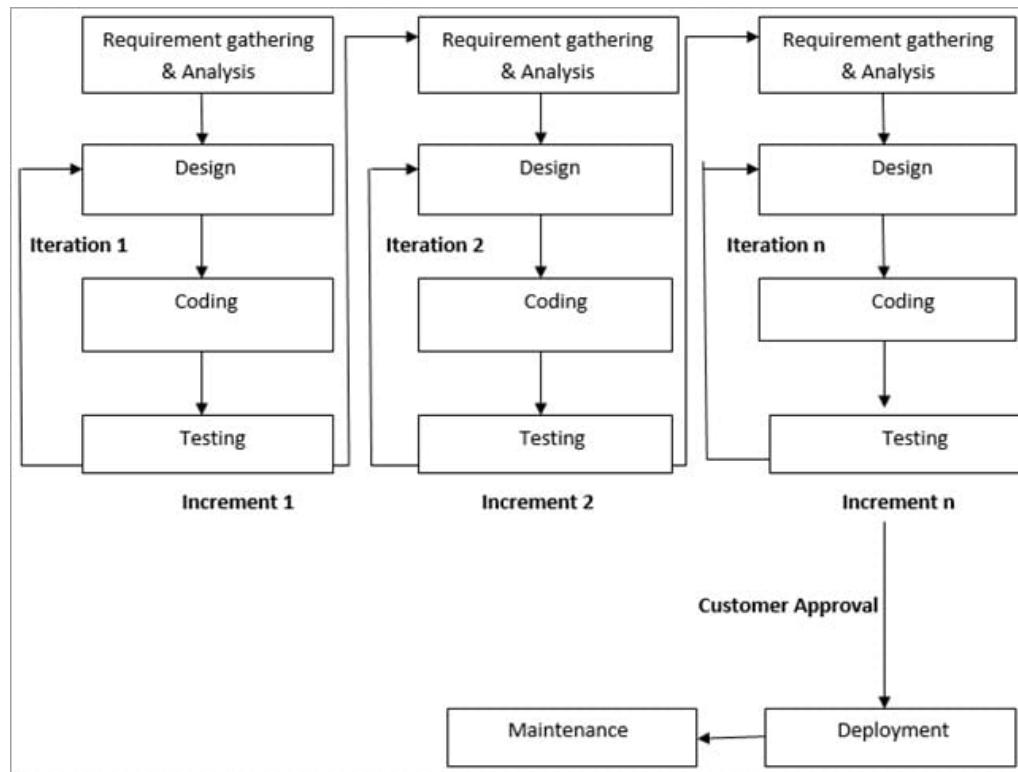


ITERATIVAS

- + Es varios ciclos de cascada



✳ MODELO ÁGIL



2

MÁS HERRAMIENTAS DE BUENAS PRÁCTICAS ITIL



* INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY

- + Guía de buenas prácticas para la gestión de servicios de tecnología.
- + Se ha elaborado para abarcar:
 - ✗ Infraestructura
 - ✗ Desarrollo
 - ✗ Operaciones



“ ITIL se enfoca en la mejora continua de los productos de software.



* PILARES DE ITIL

Procesos

Para la gestión de TI y su alineación a los objetivos de la organización.

Cliente

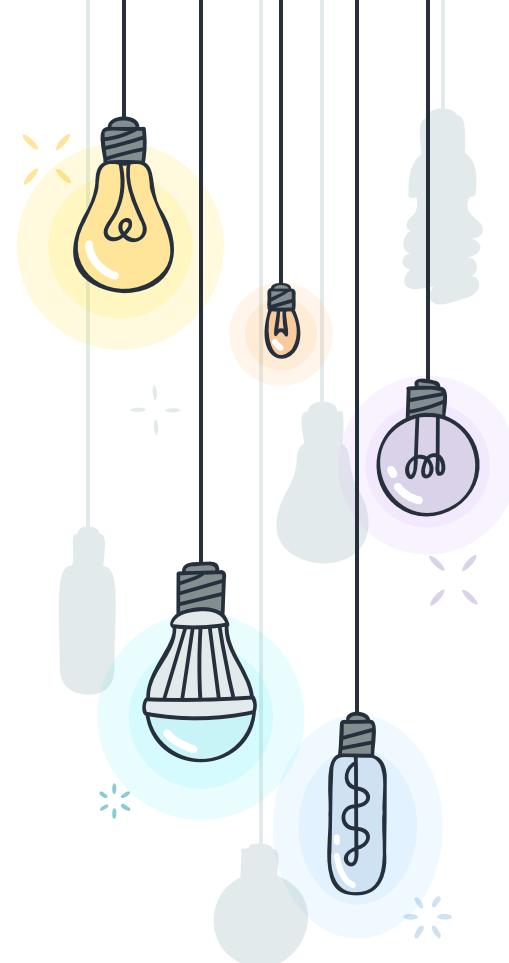
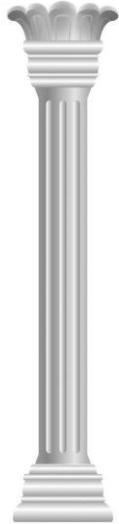
Su satisfacción es el objetivo de la mejora de los productos de TI.

Calidad

Entrega al cliente del producto o servicio óptimo, que incluya los requerimientos acordados.

Independencia

Siempre deben mantenerse buenas prácticas a pesar de los métodos para cada proceso y de los proveedores existentes.

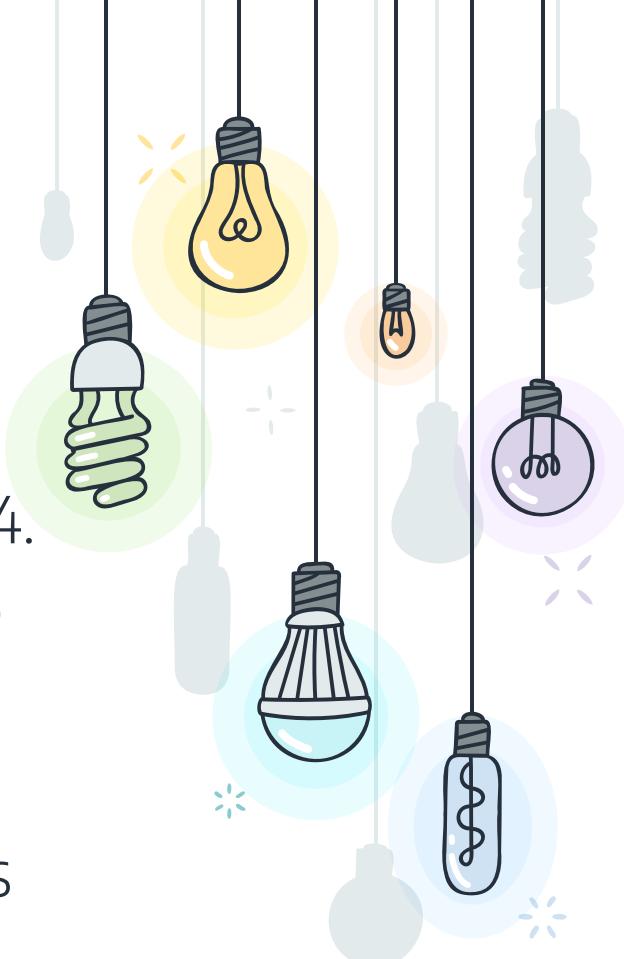


“ ITIL ha sido considerado durante mucho tiempo como la “biblia del Service Management”, es decir uno de los marcos de referencia más utilizados del mundo.



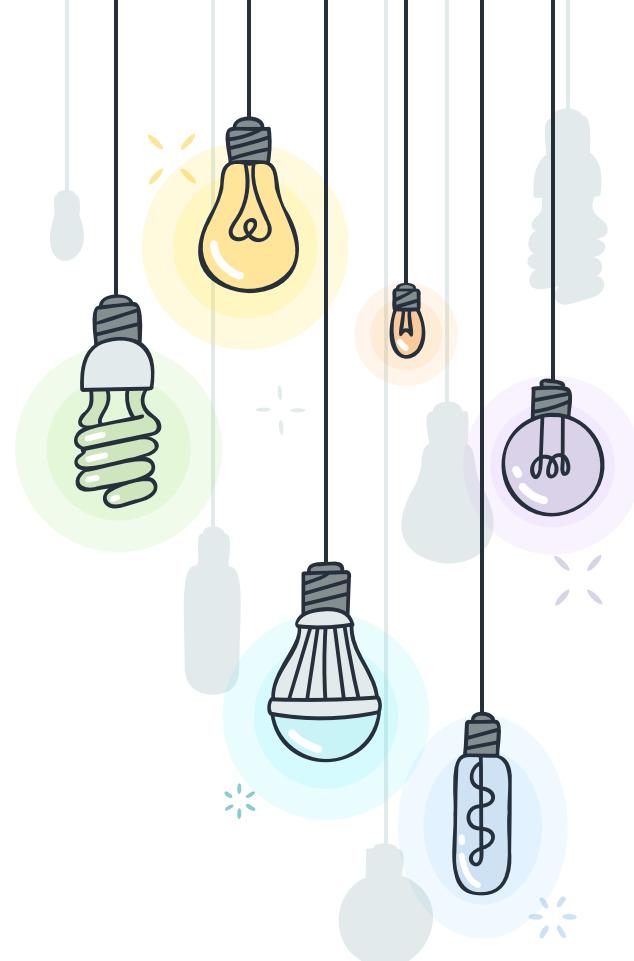
* EL PROCESO DE CREAR ITIL

- + ITIL se ha ido actualizando y ha tenido versiones.
- + 2019, se lanzó la última versión que es la 4.
- + Esta última versión ha mejorado respecto a la gestión de los servicios de TI.
- + Esto fue debido a que cada vez el reto de adaptarse a las nuevas necesidades de las organizaciones es mayor.



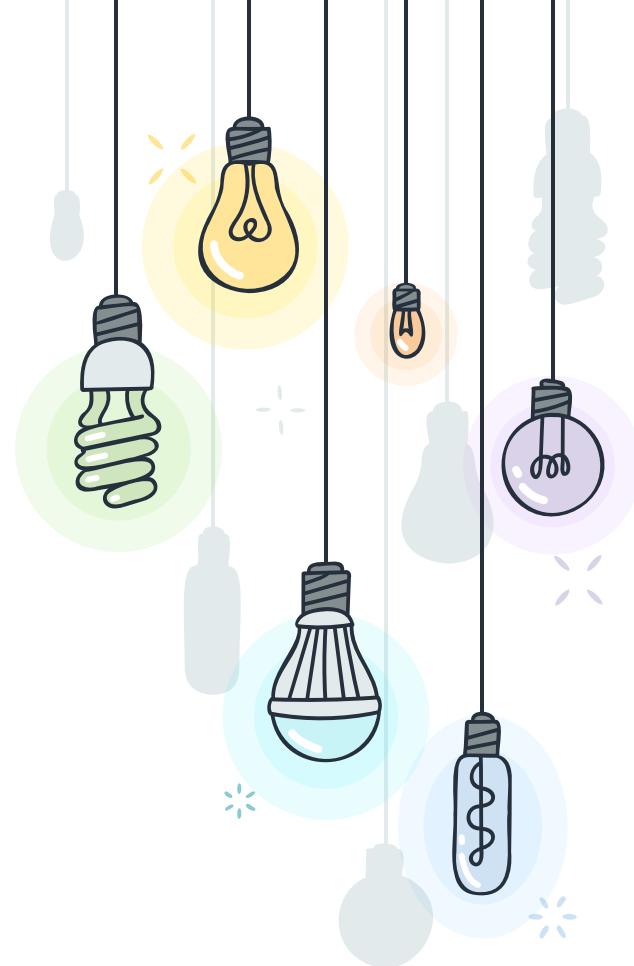
* DIMENSIONES DE LAS PRÁCTICAS DE ITIL

- + Organización y Personas
- + Información y Tecnología
- + Proveedores y Socios
- + Flujos de Valor y Procesos



ESTE
ESTE
ESTE

¿CÓMO SABE UN CLIENTE QUE SEGUIMOS LAS
MEJORES PRÁCTICAS?



* ¿EXISTEN CERTIFICACIONES DE ITIL?

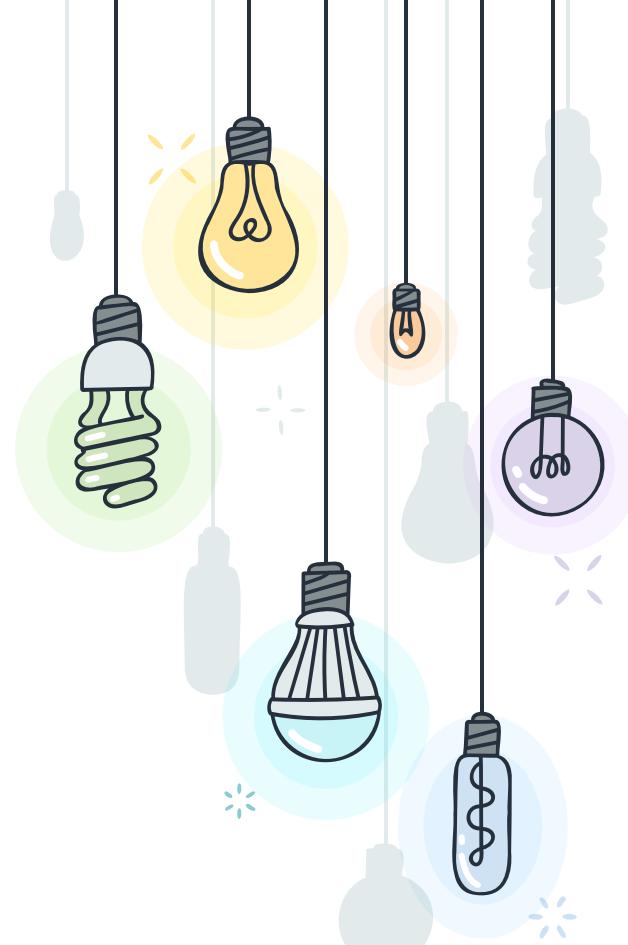
- + Sí!
- + Es la ISO 20000



“ + Podemos decir que ITIL se basa en el principio de cómo TI agrega valor a los usuarios finales o a los objetivos de un negocio



* REPASANDO LAS DIMENSIONES DE ITIL



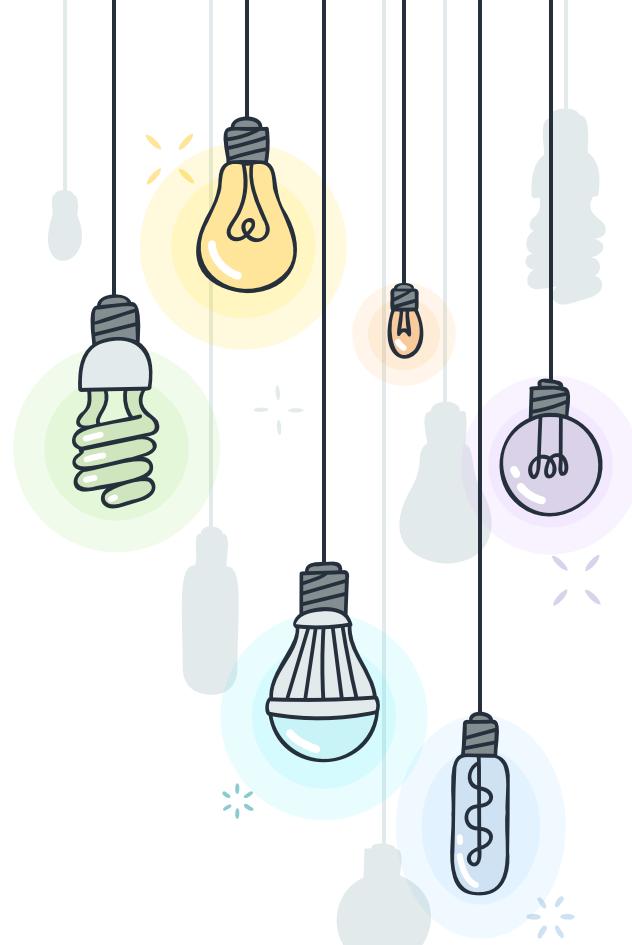
3

SISTEMA DE VALOR DEL SERVICIO ITIL



* SISTEMA DE VALOR DEL SERVICIO

- + Describe las entradas, los factores que crean valor y las salidas en la gestión de servicios de una organización.



* ENTRADAS/INPUTS

- + ¿Qué recibe nuestro sistema?
- + ¿Qué información nos dan las personas?
- + ¿Qué nos demanda el cliente o los clientes?
- + ¿Qué oportunidades tenemos?



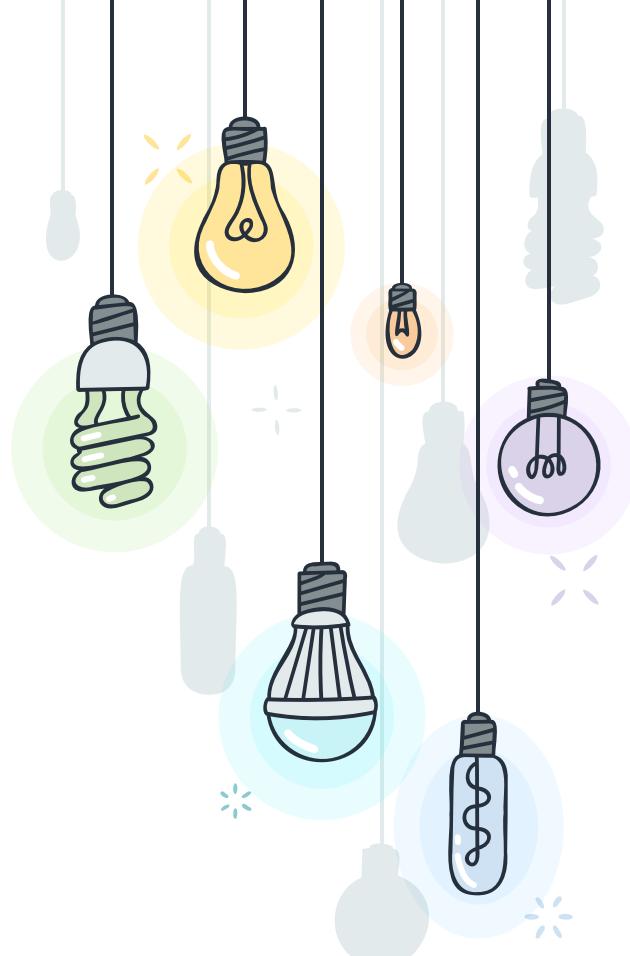
* ELEMENTOS QUE AGREGAN VALOR

- + Son todos los elementos en la organización que se relacionan o involucran de manera **directa** con la **generación de valor** para los usuarios o los objetivos del negocio.



* SALIDAS / OUTPUTS

- + Es lo que hemos terminado que genera valor
- + Está directamente relacionado con alcanzar los objetivos de una empresa o satisfacer las necesidades de los clientes.



* SISTEMA DE VALOR DEL SERVICIO

- + Busca encontrar Oportunidades potenciales de crear valor.
- + Busca también satisfacer las demandas del mercado o de los clientes.



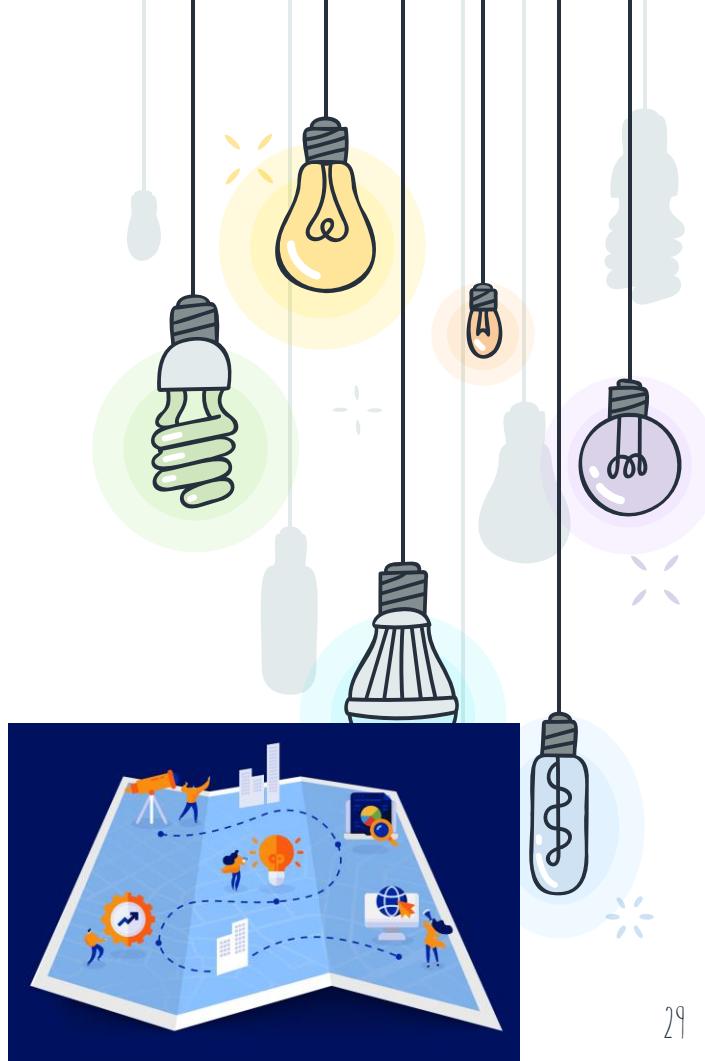
4

PRINCIPIOS GUÍA DE ITIL



* PRINCIPIOS GUÍA PARA APLICAR ITIL

- + Son un conjunto de recomendaciones
- + Tienen como objetivo guiar a la organización.
- + Guiar a pesar de diferentes cambios(organizacionales, objetivos o estrategias)



* PRINCIPIOS GUÍA PARA APLICAR ITIL

1. Enfocarse en el valor
2. Comenzar donde la empresa está actualmente
3. Progresar de forma iterativa(feedback)
4. Colaborar y promover visibilidad
5. Pensar y trabajar holísticamente
6. Mantenerlo simple y práctico
7. Optimizar y Automatizar



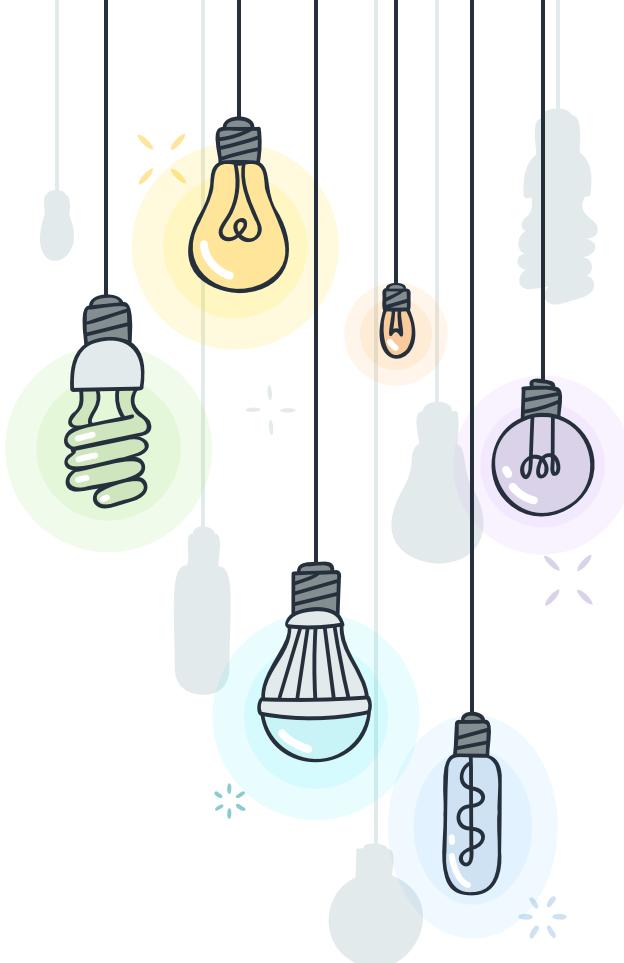
5

CADENA DE VALOR DE SERVICIOS ITIL



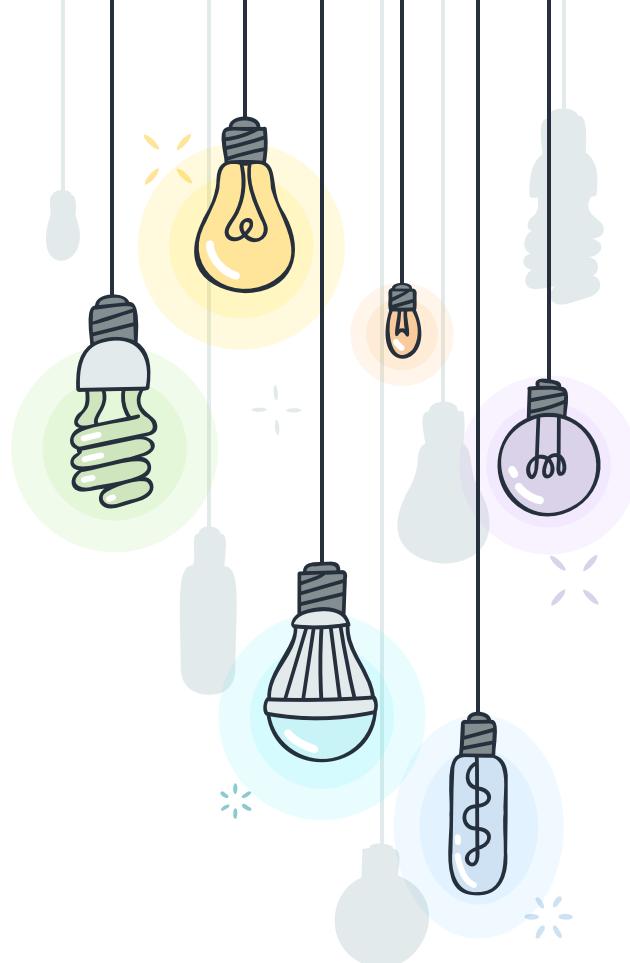
* CADENA DE VALOR DEL SERVICIO

- + Def. Es una combinación de 6 actividades que trabajan en conjunto creando valor para la organización o los usuarios, a través de la entrega de un producto o servicio.



* CADENA DE VALOR DEL SERVICIO

1. Planificación
2. Mejora
3. Compromiso
4. Diseño y Transición
5. Obtener
6. Entrega
7. Soporte



6

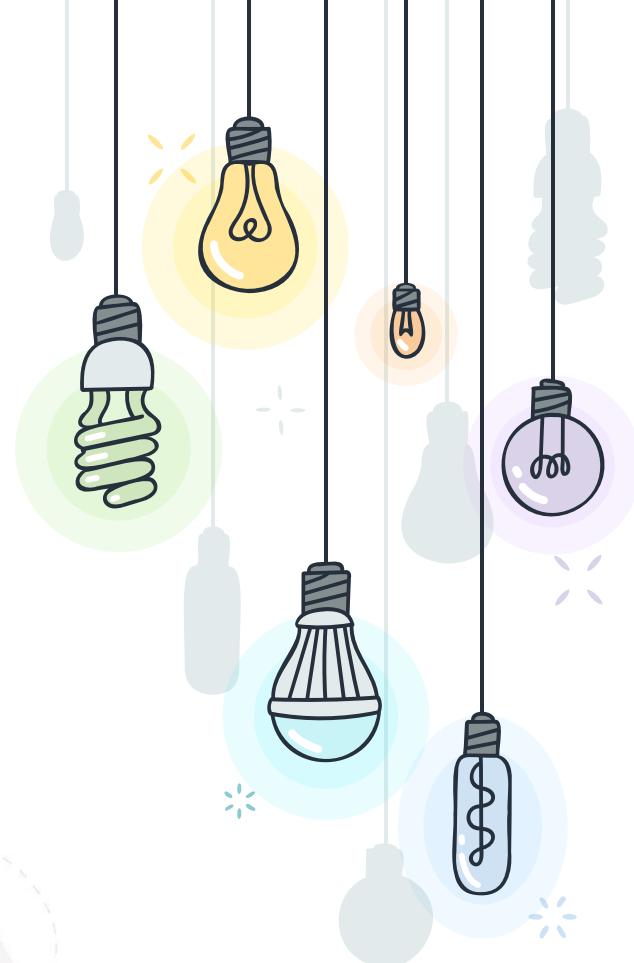
GOBIERNO DE ITIL

ITIL



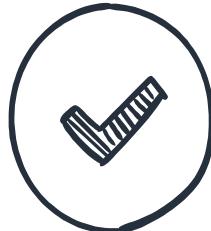
* GOBIERNO DE ITIL

- + Evaluación
- + Dirección
- + Supervisión

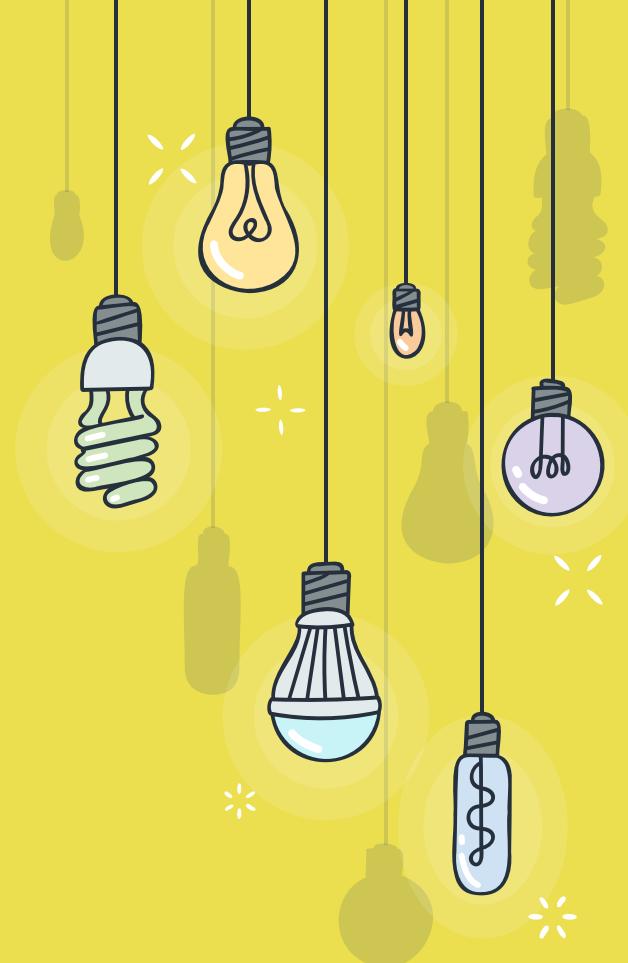


* OBJETIVO DEL GOBIERNO DE ITIL

- + Garantizar que la cadena de valor del servicio y las prácticas de la organización funcionen alineadas con los objetivos de la empresa.



“ + La alta calidad y mejora
Continua son la esencia
para agregar valor.



7

PRÁCTICAS DE ITIL



* PRÁCTICAS DE ITIL



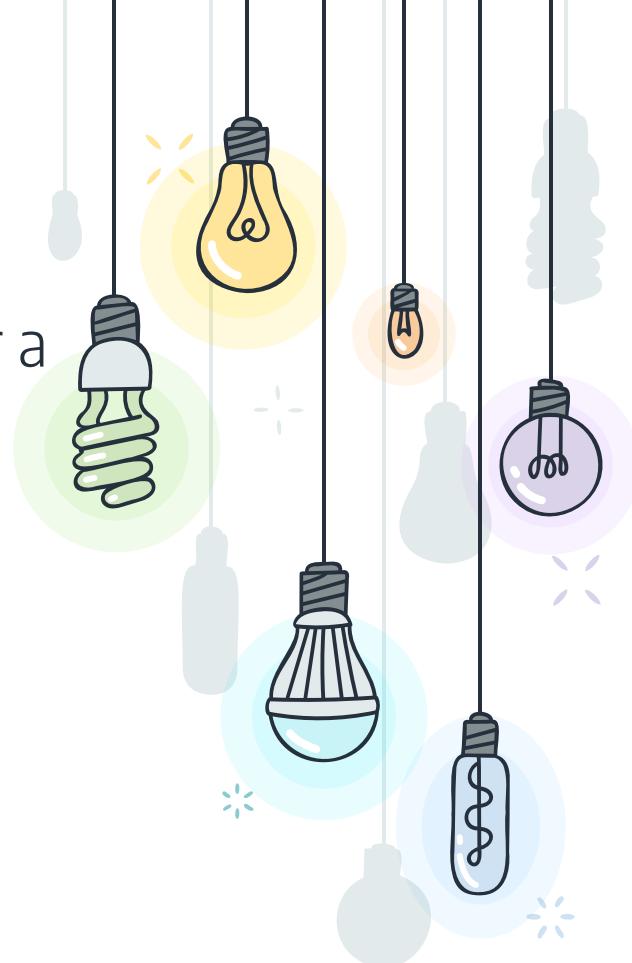
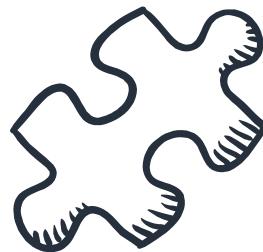
8

REPASO E INTEGRACIÓN DE... ITIL



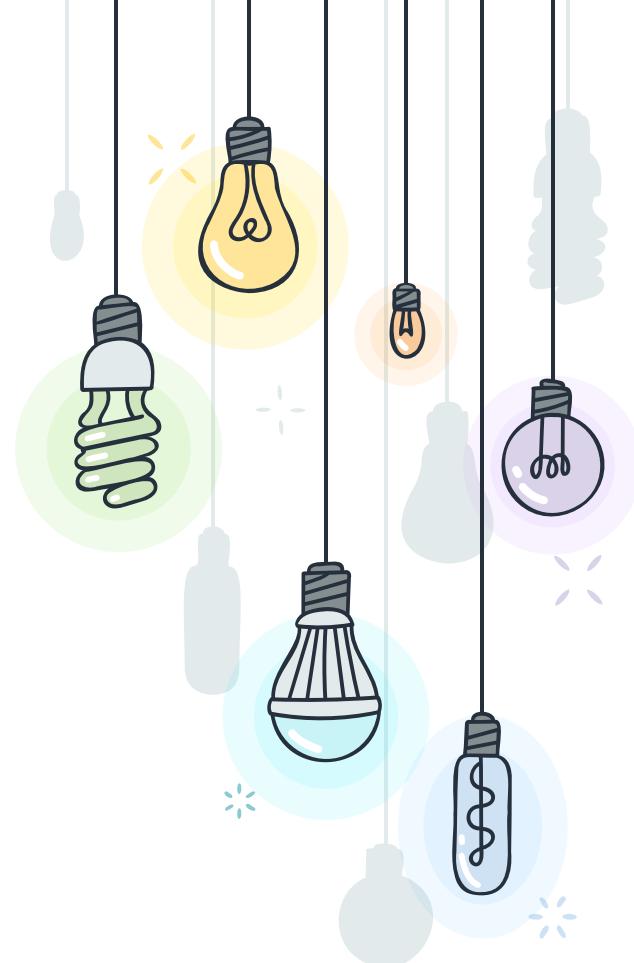
* CONCLUSIONES SOBRE ITIL

- + **Servicio:** Es un medio para entregar valor a los clientes.
- + Gestionar el servicio
- + Cadena de valor del servicio

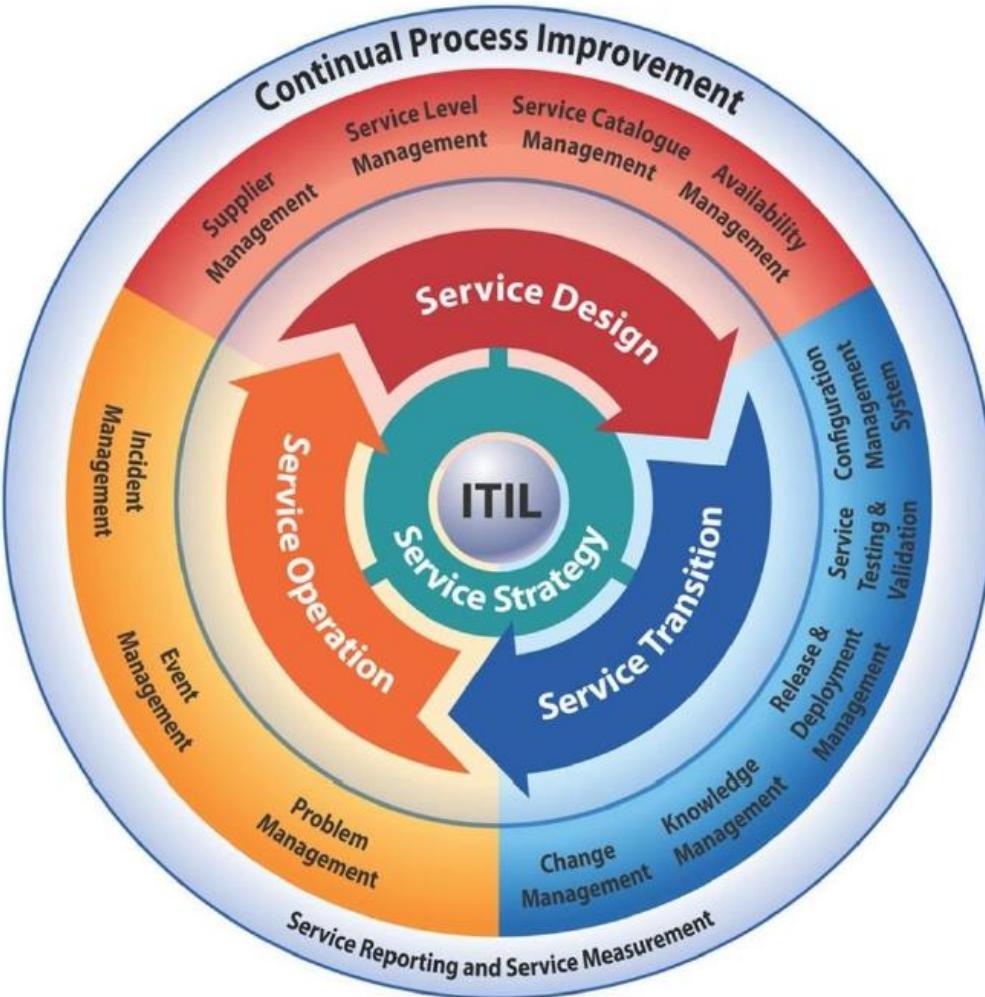


* PODEMOS DECIR DE ITIL QUE:

- + No es propietario
- + Consiste en las mejores prácticas
- + Se centra en las 4 áreas o dimensiones organizacionales para agregar valor

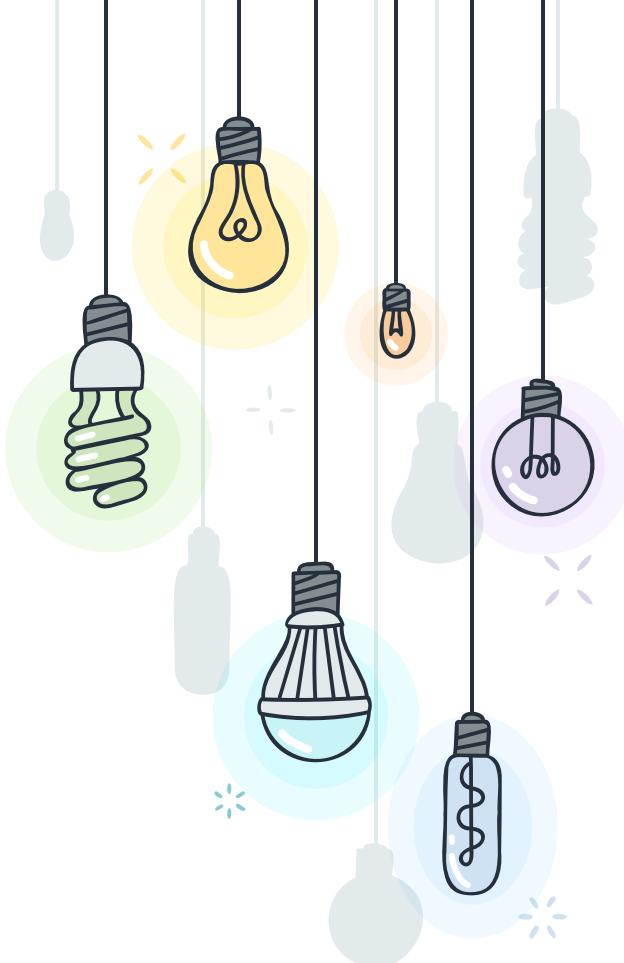


* EL PROCESO DE ITIL



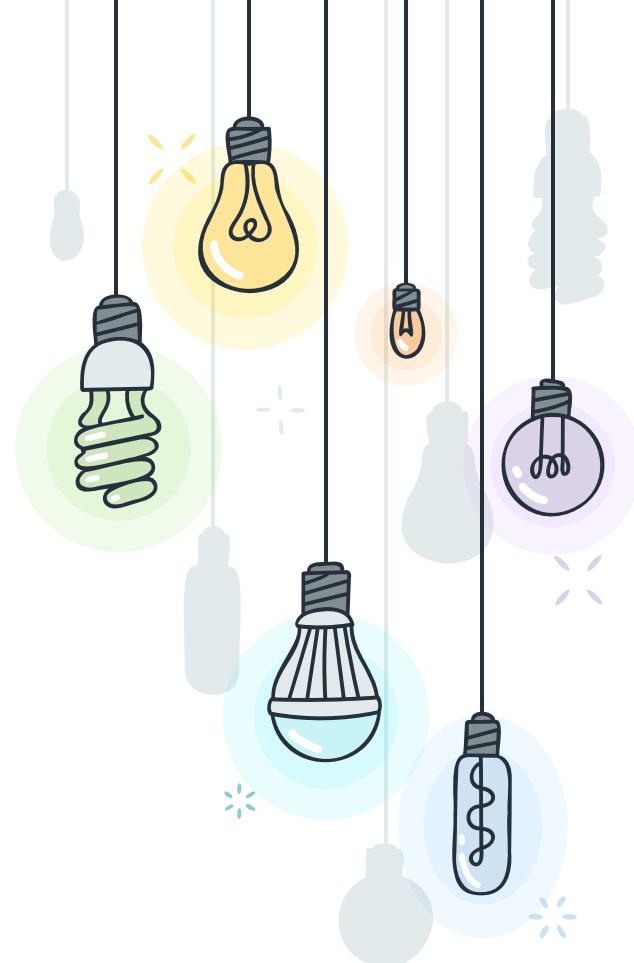
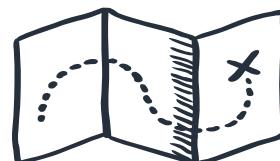
* SEGÚN EL DIAGRAMA ANTERIOR

- + 5 grandes secciones o “libros”
- + ITIL Service-Strategy
- + **ITIL Service-Design**
- + ITIL Service Transition
- + **ITIL Service Operation**
- + ITIL Continual Service Improvement



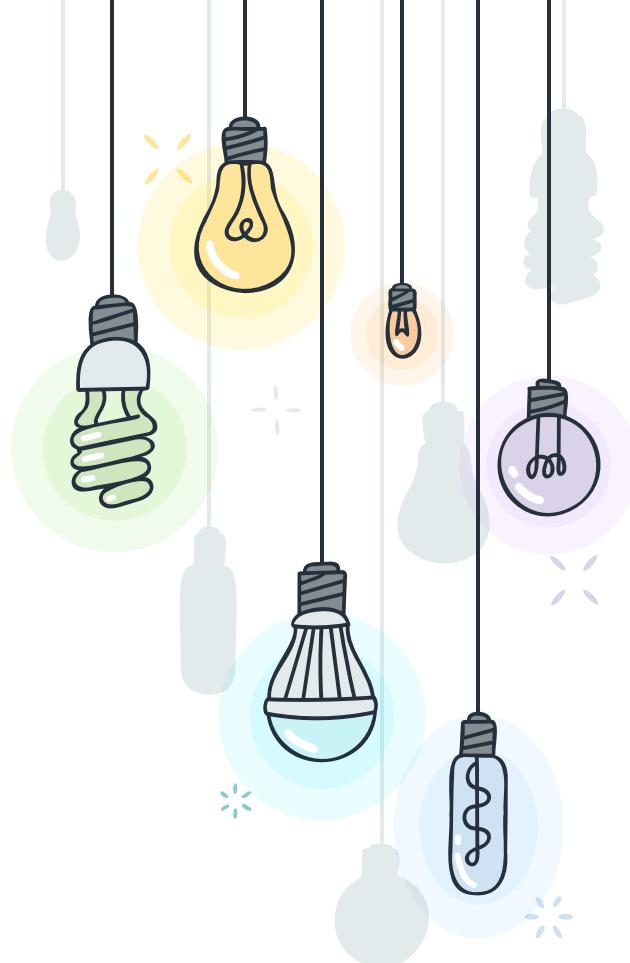
* ESTRATEGIA DE SERVICIO

- + Gestión financiera
- + Análisis de demanda y su gestión
- + Gestión del portafolio de servicios

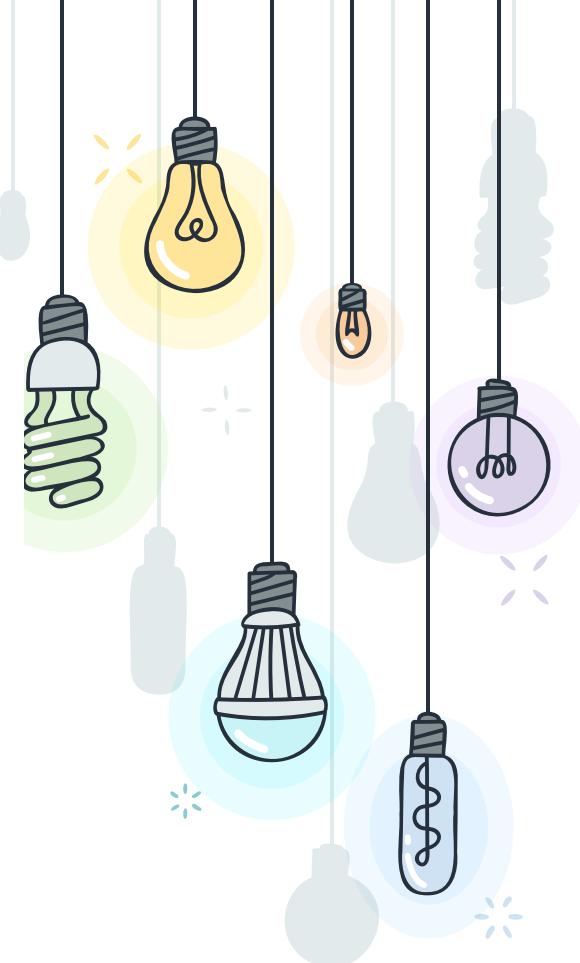
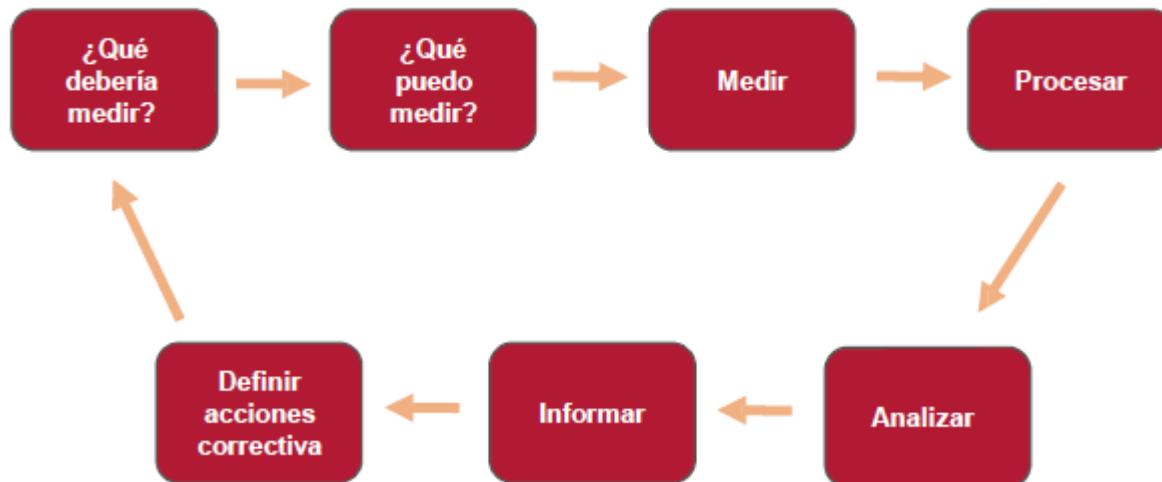


✳ MEJORA CONTINUA

- + Dadas las mediciones realizadas, identificar oportunidades de mejoras o implementar acciones correctivas.

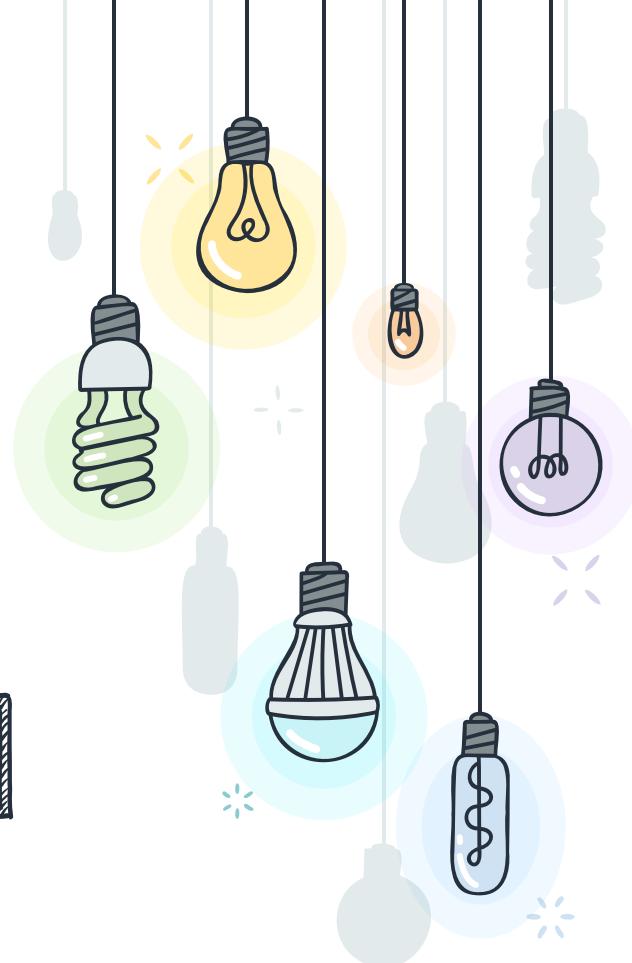


MEJORA CONTINUA

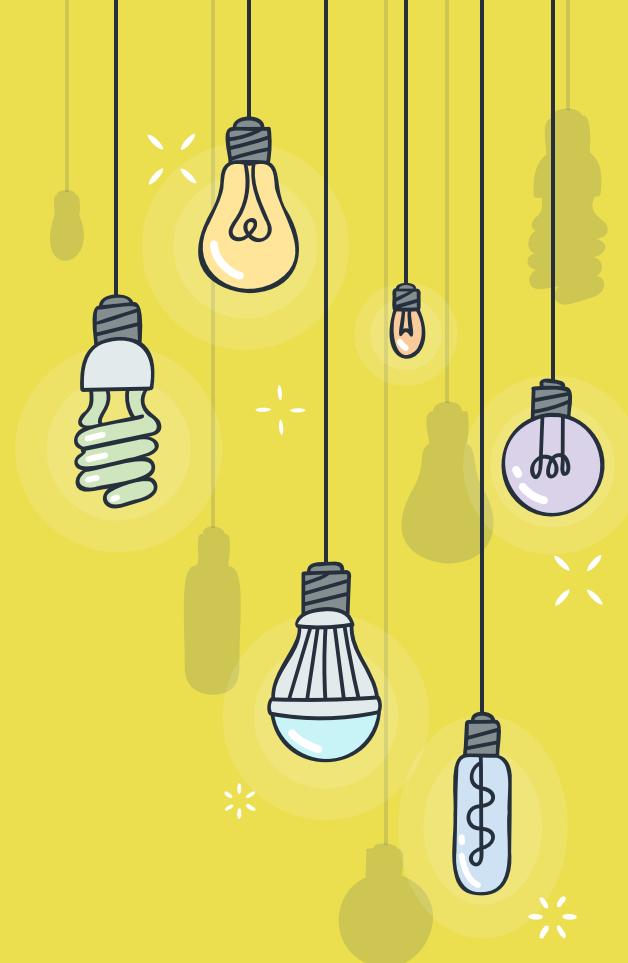


* HABLEMOS DE MÉTRICAS

- + Tecnológicas
- + De procesos
- + De servicios

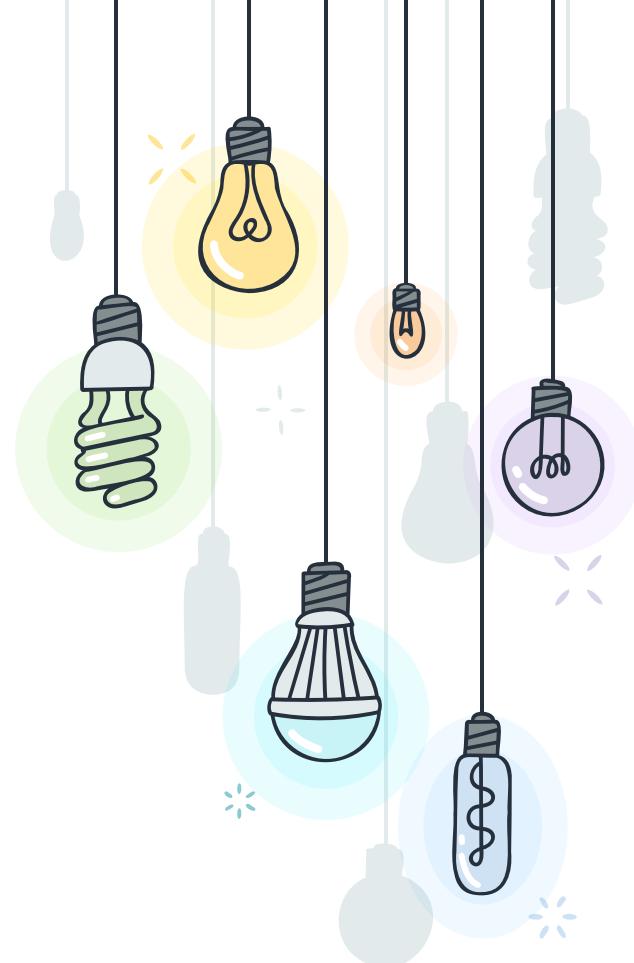


“ + Repasando los 3 libros de ITIL ya vistos en clase



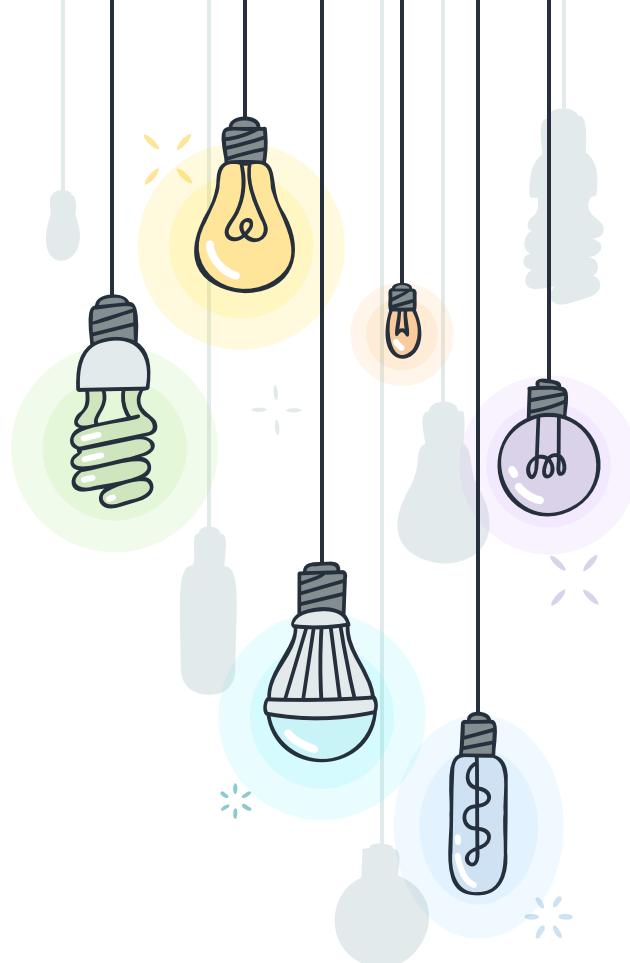
* DISEÑO DEL SERVICIO

- + Gestionar catálogo de servicios
- + Gestionar niveles de servicio
- + Gestionar capacidad
- + Gestionar Disponibilidad
- + Gestionar Continuidad
- + Gestionar Seguridad
- + Gestionar proveedores



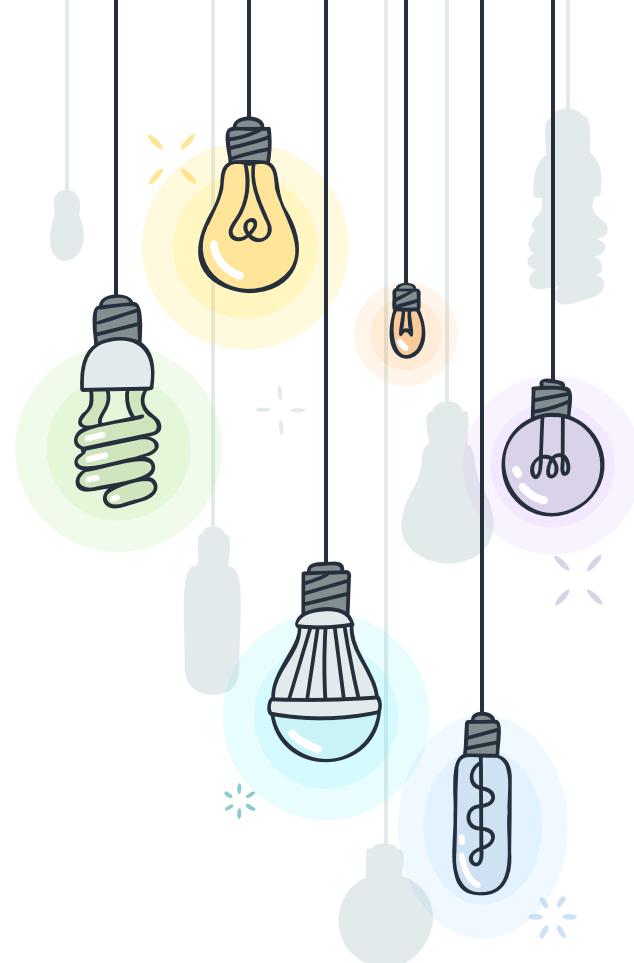
* TRANSICIÓN DE SERVICIO

- + Gestionar los procesos, sistemas y funciones requeridas para construir, probar y desplegar un servicio a un ambiente de producción



* TRANSICIÓN DEL SERVICIO

- + Gestión de cambios
- + Gestión de activos y configuraciones
- + Gestión del conocimiento
- + Planificación y control
- + Gestión de versiones y entregas
- + Validación
- + Evaluación



* OPERACIÓN DEL SERVICIO

- + Gestión de eventos
- + Gestión de Incidentes
- + Gestión de problemas
- + Gestión de peticiones
- + Gestión de accesos
- + Gestión técnica
- + Gestión de aplicaciones

