Contenido

[Arquitectura de Software 2](#_Toc88519447)

[ARQUITECTURA: nos habla de una abstracción 2](#_Toc88519448)

[¿Por qué la arquitectura de Software es importante? 3](#_Toc88519449)

[“¿Cuáles son los interesados de una Arquitectura?” 3](#_Toc88519450)

[Atributos de Calidad 4](#_Toc88519451)

[Atributos de Calidad – Disponibilidad 4](#_Toc88519452)

[Atributos de Calidad – Interoperabilidad 4](#_Toc88519453)

[Atributos de Calidad – Adaptabilidad 5](#_Toc88519454)

[Atributos de Calidad - Performance 5](#_Toc88519455)

[Atributos de Calidad – Seguridad 5](#_Toc88519456)

[Atributos de Calidad – Usabilidad 5](#_Toc88519457)

[Atributos de Calidad – Capacidad de Prueba y Testeo 5](#_Toc88519458)

[OTROS ATRIBUTOS DE CALIDAD 6](#_Toc88519459)

[Tácticas de Arquitectura y Patrones 6](#_Toc88519460)

[Un patrón de Arquitectura: 6](#_Toc88519461)

[Los ASR en la Arquitectura… 7](#_Toc88519462)

[La Arquitectura en los Proyectos ágiles 7](#_Toc88519463)

[Evaluación de una Arquitectura 7](#_Toc88519464)

[Arquitectura Cloud (acá importa la integración de las arquitecturas) 8](#_Toc88519465)

[Arquitectura Cloud ( depende de que recursos comparten) 8](#_Toc88519466)

[Arquitectura en un entorno “Cloud” 8](#_Toc88519467)

[“Los atributos de calidad que tienen diferencias, son: 8](#_Toc88519468)

[MODELO C4 o de las 4Cs 9](#_Toc88519469)

[Nivel 1: 10](#_Toc88519470)

[Nivel 2: 11](#_Toc88519471)

[Nivel 3 o de los componentes 12](#_Toc88519472)

[Y el nivel 4 o cuando llegamos finalmente el código: 13](#_Toc88519473)

[Consultas 13](#_Toc88519474)

[ANEXO: 14](#_Toc88519475)

[¿Qué es la interoperabilidad? 14](#_Toc88519476)

**Desde el área de gestión de IT, como puede ser gerentes de IT o colaboradores de gerencia de IT, y parte de ese trabajo es administrar muchas aplicaciones, cantidad de 1,2 o 3 dígitos, pero el que administra enfrenta la cantidad, si algún escritorio se cae entonces podríamos no enterarnos a tiempo, entonces si no hay buen sistema de información entonces se puede tener consecuencias negativas en la gestión**

**Esas aplicaciones que gestionamos, deben de interactuar entre sí, pero dos apps distintas, como “mundos distintos” o “mundos separados”, algunos datos tendientes de una solución puede haber diferencias significativas, por ejemplo, el 1,8 y 1,7 m de altura de las personas. Luego esta visión de la arquitectura (como es por dentro la app), es decir de la forma en que armamos el software puede ayudar armar el SW, como interactúa entre sí, y cómo funciona la misma, pero la arq no resuelve el vivir en un “vecindario”, pero a partir de la aceptación de las existencias de otras soluciones que estamos dispuestos a poder controlar. Una Arquitectura solida resuelve muchos problemas.**

# Arquitectura de Software

► Definición**: es una forma de abstracción de la forma en que se construye un software, algo así como el esqueleto de una solución, pero ese esqueleto se referiría mirar todas las aplicaciones, una especie de meta arquitectura, de las arquitecturas ahí volcadas**

“Define de manera abstracta, el conjunto de estructuras que la componen. Son elementos de **tecnología**, **relaciones** y **propiedades** entre ellas”.

► Objetivos**:**

“Los sistemas de software son construidos para satisfacer los objetivos del negocio”.

► En que consiste:

**Estructuras => Elementos => Relaciones**

“La Arq. omite ciertos detalles internos de cada elemento, se abstrae de su dificultad, y se ocupa de lo exterior”.

**Interfaces: -** Dividen lo privado de lo público, se centra en la complejidad de la interacción de los elementos.

ARQUITECTURA: nos habla de una abstracción de la forma que se construye un software, un esqueleto, se referiría mirar a todas las aplicaciones, se mira las arquitecturas de productos ya existentes, se mira un conjunto de soluciones para problemas puntuales

No podemos vivir razonablemente bien si no pensamos en el concepto de arquitectura

**“NO TODAS LAS ARQUITECTURAS SON BUENAS”**

# ¿Por qué la arquitectura de Software es importante?

► Usuario dependiente de la rapidez, disponibilidad y confiabilidad de los sistemas.

► Cliente preocupado porque se implemente una arquitectura, bajo calendario y presupuesto seleccionado.

► Project Manager, preocupado porque los equipos trabajen en forma independiente interactuando con disciplina.

► El Arquitecto se ocupa que los 3 puntos antes mencionados funcionen correctamente y en forma sincronizada

Se pregunta siempre que se genera con la arquitectura

**Arquitectura de Software**

# “¿Cuáles son los interesados de una Arquitectura?”

► Clientes.

► Usuarios.

► Project Manager.

► Arquitecto.

► Desarrolladores.

► Testers.

► Y más…

Donde afecta las decisiones arquitectonicas afectan y de esa manera conocer sus intereses, algunos de esos roles o interesados no técnicos o participan de forma directa, pero si hay una participación, no olvidar de nuestra área de IT de staff por que puede haber otro que nos necesita y no estamos por nosotros mismos.

En la arquitectura hay que tener en cuenta aspectos técnicos y de negocios entre otros.



Cuando uno quiere medir la calidad de la arquitectura se hace a través de atributos de calidad, arquitectura de cualquier tipo, hay que tener en cuenta:

# Atributos de Calidad

Es una propiedad de medida o de testeo que permite indicar que tan bien funciona un sistema y como satisfacer las necesidades de los interesados.

► Requerimientos Funcionales: lo que la arquitectura permite, la funcionalidad

► Requerimientos de calidad del sistema

► Restricciones

Atributos de Calidad – Disponibilidad **(la arquitectura aporta mucho de este aspecto, habla que porcentaje del tiempo acordado o servicio o componente de un servicio estuvo ahí para ser utilizado, ejemplo: tiempo disponible de cola para la gente, poder asegurar la disponibilidad está ligado a la arquitectura de la solución)**

1. Minimizar las interrupciones del servicio

2. Mitigar posibles fallas que puedan ocurrir (que as su vez genera esas interrupciones)

Tácticas:

► Detectar Fallas

► Recuperación de Fallas

► Prevención de Fallas

Atributos de Calidad – Interoperabilidad **(habla de la posibilidad de intercambiar información, es decir las aplicaciones interactúan, esto es un elemento central, es intercambio de información para accionar como información de control, por ejemplo)**

1. 2 o más sistemas pueden intercambiar información vía interfaces y hasta comprender dicha información.

2. Si conocemos las interfaces de los sistemas externos, donde nuestros sistemas operan, podemos diseñar este conocimiento.

Tácticas:

► Locate – Los sistemas que operan deben ser descubiertos en tiempo de ejecución.

► Manage Interfaces – Agrega o elimina capacidades de una interface.

Atributos de Calidad – Adaptabilidad **(cuando a la solución resuelva los problemas actuales y los futuros que no hemos imaginado, la necesidad de flexibilidad en la arquitectura y estructura, hay que tener una estructura sólida pero que se pueda cambiar, pero por que la calidad cambia)**

► Cambio

► Costo (cuando es esfuerzo para lograr se sujeta a los costos altísimos)

► Riesgo

## Atributos de Calidad - Performance

►Tiempo

►Habilidad

Atributos de Calidad – Seguridad **(la arquitectura puede aportar a la seguridad, puede incluir aspectos de seguridad)**

► Detectar Ataques

► Detección de Intrusos

► Denegación de un servicio

► Verificación de integridad de msj

► Atraso en los mensajes

► Resistir Ataques

► Autenticación de actores

► Límite de acceso

► Encriptación de Datos

## Atributos de Calidad – Usabilidad

► “Cuán fácil es para el usuario ejecutar una tarea deseada”

► “Es una de las formas más fáciles de mejorar la calidad de un sistema”

Atributos de Calidad – Capacidad de Prueba y Testeo **(la arquitectura como elemento del software debe ser probado, que funcione y cumple sus objetivos, los requerimientos deben ser verificables entonces que la solución cumpla un requisito, entonces el chequear incluye a lo que es la arquitectura, recordemos que el requerimiento debe cubrir una necesidad, entonces la arquitectura tiene ciertos objetivos y debemos verificar que se cumpla estos objetivos, el costo no importa sino que se pueda cumplir los requisitos)**

* **“Entre el 30 y el 50% del costo de una buena ingeniería en el desarrollo de los sistemas es absorbida por las pruebas”**

## OTROS ATRIBUTOS DE CALIDAD

► Variabilidad **= Adaptación al contexto.**

► Portabilidad **= Cambios de plataforma.**

► Desarrollo Distribuído **= Diseño del Soft.**

► Escalabilidad **= Agregar mas recursos, como un server.**

► Monitoreo **= Investigar el sistema mientras trabaja.**

► Comerciabilidad **= No siempre se adapta a lo que necesitamos.**

Tácticas de Arquitectura y Patrones **(similar a patrón de diseño, es decir tener soluciones a problemas complejos y para ciertos problemas aparecen ciertas soluciones)**

“Tener éxito en el diseño de la arquitectura es complejo y cambiante, por eso los diseñadores buscaron las mejores formas de reutilizar el conocimiento arquitectónico”

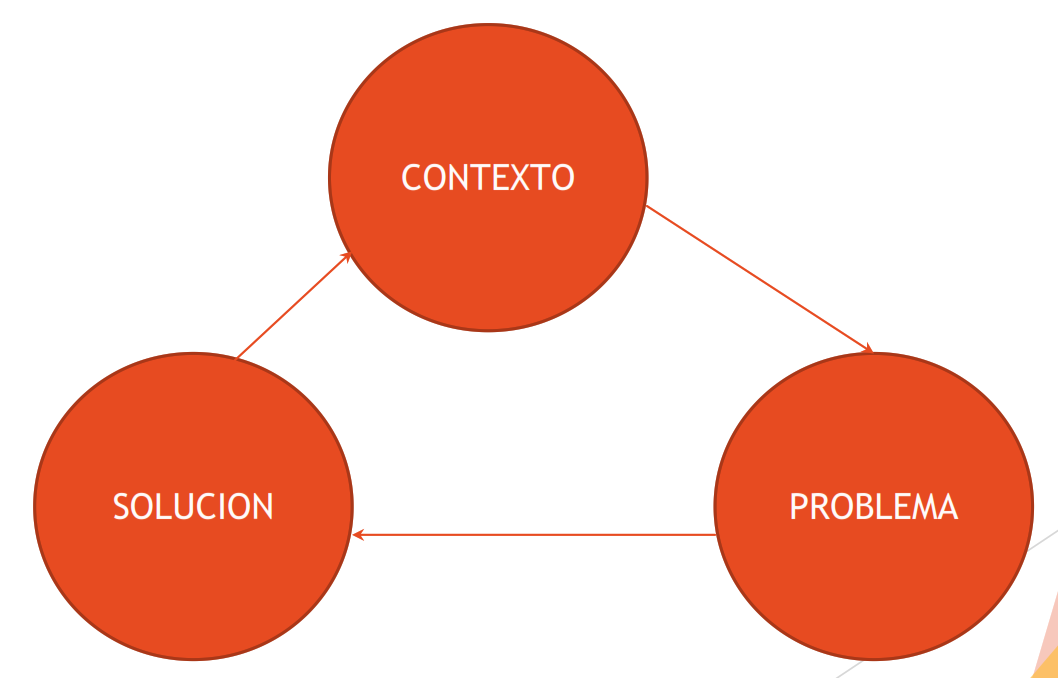
# Un patrón de Arquitectura:

► Es un paquete de decisiones de diseño que se encuentra en la práctica.

► Conoce propiedades que permiten reutilización.

► Describe un class de arquitecturas.

PATRONES DE ARQUITECTURA: Relación entre …



Para construir una arquitectura se requieren requerimientos, entonces la arquitectura se construye a partir de ciertos requerimientos arquitectónicos

ASR (Requerimientos de Arquitectura Significativos) en los Ciclos de Vida

## Los ASR en la Arquitectura…

► Reunir los ASRs de los documentos de requerimientos

► Reunir los ASRs entrevistando a los interesados

► Reunir los ASRs entendiendo los objetivos del Negocio

# La Arquitectura en los Proyectos ágiles

“Los métodos y procesos se han agilizado y los proyectos han tenido que cambiar”

**Puntos importantes:**

1. Alta satisfacción del cliente cuando se entrega una versión.

2. Si cambian los requerimientos, aunque sea tarde, se adapta bien.

3. Entregas de software: Entre semanas y meses, tiempos en general cortos.

4. Hay gran interacción entre la gente del negocio, y la gente de IT. Interacción cara a cara.

5. Motivación del grupo de trabajo.

“En muchas organizaciones existe una **combinación** de Arquitecturas que se basan en tipos de **proyectos ágiles** y Arquitecturas de paradigmas **estructurados**, no siempre hay que caer en lo que ofrece el mercado como solución.” Hay combinaciones de enfoques en un mismo proyecto o algo similar, las arquitecturas suelen ser una combinación de enfoques, y cada enfoque caracteriza a la arquitectura. (No vendemos herramientas, vendemos soluciones a los problemas)

Gestión y Gobierno: “reglas en las relaciones del vecindario” son las relaciones de negociaciones, cuales son las ideas personalizables,

# Evaluación de una Arquitectura

1. El Arquitecto debe interesarse por la gestión de proyectos…

2. El PM es la persona que, junto al Arquitecto, deben trabajar en conjunto, por la perspectiva de la organización…

3. A Mayor complejidad de proyectos más útil es la implementación de una arquitectura…

**(a veces las fallas están en no articular bien los intereses de todos)**

Planificación

1. La planificación de un proyecto sucede constantemente, pero existe un plan inicial para convencer a la dirección de construir el sistema y dar una idea de costo y agenda…

2. El PM debe educar a otros managers para poder corregir desvíos en el desarrollo del software…

Organización

► Team Leader – Gestiona las tareas del equipo

► Developer – Diseñan e implementan los subsistemas de código.

► Configuration Manager – Ejecutan y construyen test de integración.

► System Test Manager – Testeo de sistema y testing de aceptación.

► Product Manager – Representan el marketing.

# Arquitectura Cloud (acá importa la integración de las arquitecturas)

Las arquitecturas onpremise y cloud pueden interactuar entre si para brindar varios servicios, entonces las soluciones deben tener una arquitectura para integrar a una arquitectura

► Servicios a Demanda: la oferta acompaña rápidamente a la demanda

► Acceso único de red: los servicios se consumen a través de una red que costa de un pool de recursos

► Pool de recursos: en general esto no se suele ver, sino el servicio mismo

► Independencia de ubicación: no es absoluto, por que hay pedidos que ofrecen ubicación donde se puede usar los servicios cloud por ejemplo, pero en términos generales esto no está especificado

► Elasticidad rápida: si la oferta se acomoda rápidamente a la demanda, a veces de forma no automática (si viene mucha gente debemos atender, hay que ordenar, nadie lo hace de forma automática)

► Servicios medidos: por que el enfoque es pagar por usar, según algún criterio luego se factura

## Arquitectura Cloud ( depende de que recursos comparten)

“Los modelos de desarrollo del cloud se diferencian por quienes son dueños y quienes lo operan”.

Existen 2 modelos básicos que tienen 2 variantes: depende de que recursos se comparten

► Cloud Privado: el mismo que ofrece el servicio publico tiene opciones privadas con servicios exclusivos, lo que compartimos con otros va o está por otro lado, entonces tenemos garantizado el servicio detrás se ese servicio cloud

► Cloud Publico: si para una organización los recursos son mandatorios o muy importante

## Arquitectura en un entorno “Cloud”

“El Arquitecto necesita prestar atención en la adaptabilidad, la usabilidad, la interoperabilidad y el testeo, como haría en otra plataforma”.

## “Los atributos de calidad que tienen diferencias, son:

► Seguridad: normativas propias de la organización, o legales, que permite los servicios cloud o los restrinja, o que lo que va en cloud va en algunas formas como en donde va a residir la propia arquitectura. También con aspectos de confidencialidad, que puedan estar conectado directamente con el proveedor, o la ubicación de la infraestructura cloud

► Performance: como por ejemplo la conexión al servicio cloud

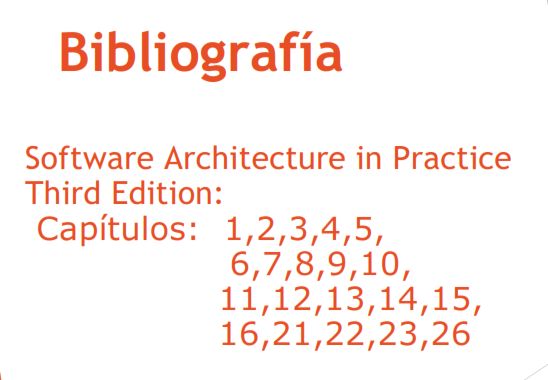
► Disponibilidad: dependemos del proveedor, pero no todo el proveedor por ejemplo tendremos problema de red y el proveedor cumplió todo, algunos proveedores de cloud ofrecen la propia conectividad hacia ellos

Bibliografía:

* Software Architecture in Practice: Software Architect Practice\_c3 (SEI Series in Software Engineering) 3rd Edition. Len Bass, Paul Clements, Rick Kazman.
* <https://c4model.com>
* <https://www.linkedin.com/pulse/una-buena-arquitectura-facilita-la-agilidad-gabriel-monreal/>

Arquitectura y agilidad:

* <http://johanachuquino.com/la-agilidad-vision-del-arquitecto-software/>   
  <https://www.linkedin.com/pulse/una-buena-arquitectura-facilita-la-agilidad-gabriel-monreal/>   
  <https://ieeexplore.ieee.org/abstract/document/5420791>   
  <https://www.computer.org/csdl/magazine/so/2009/05/mso2009050002/13rRUxASutt>



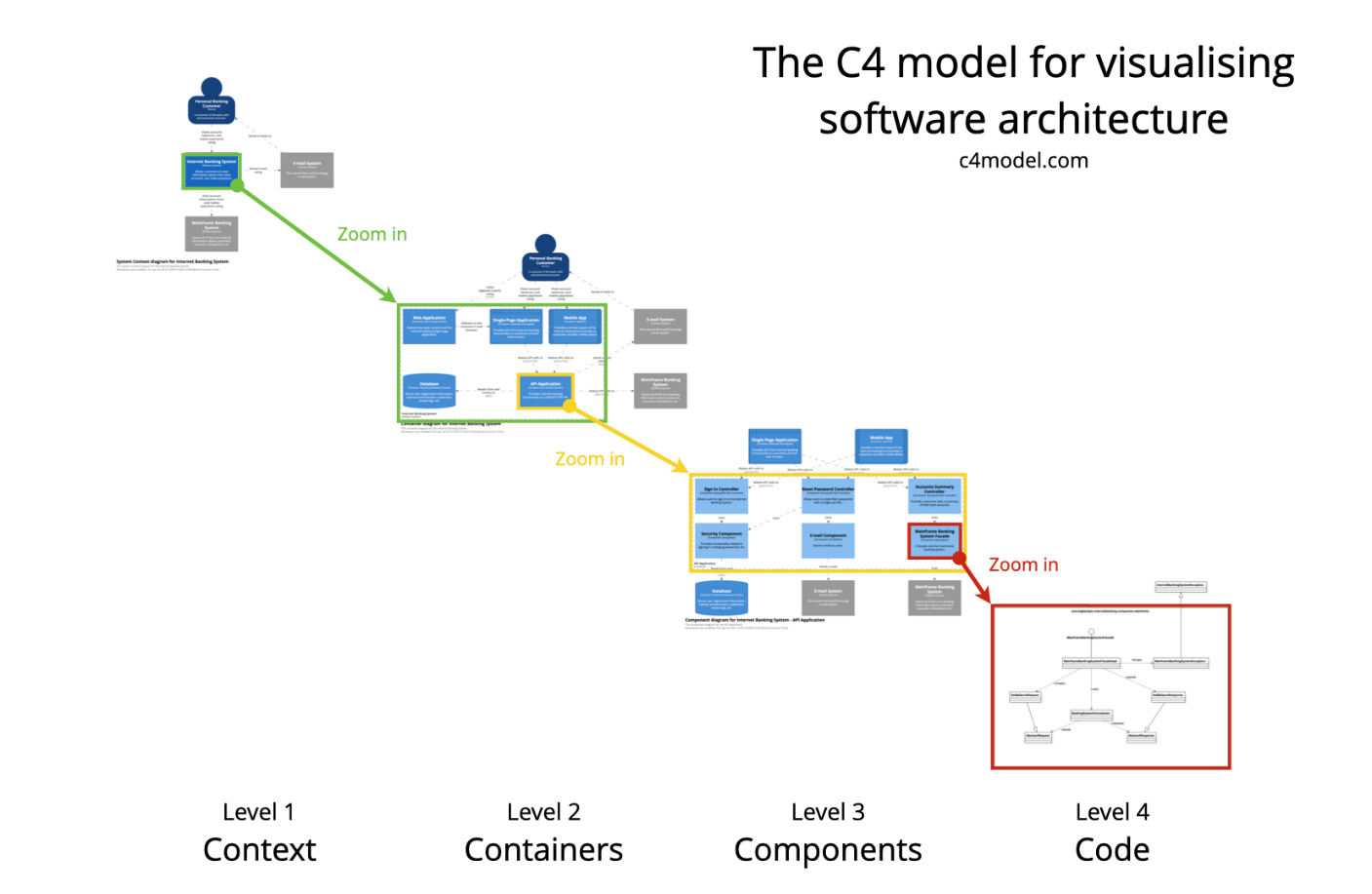
# MODELO C4 o de las 4Cs

Nivel 1: contexto;

Nivel 2: contenedores;

Nivel 3: componentes;

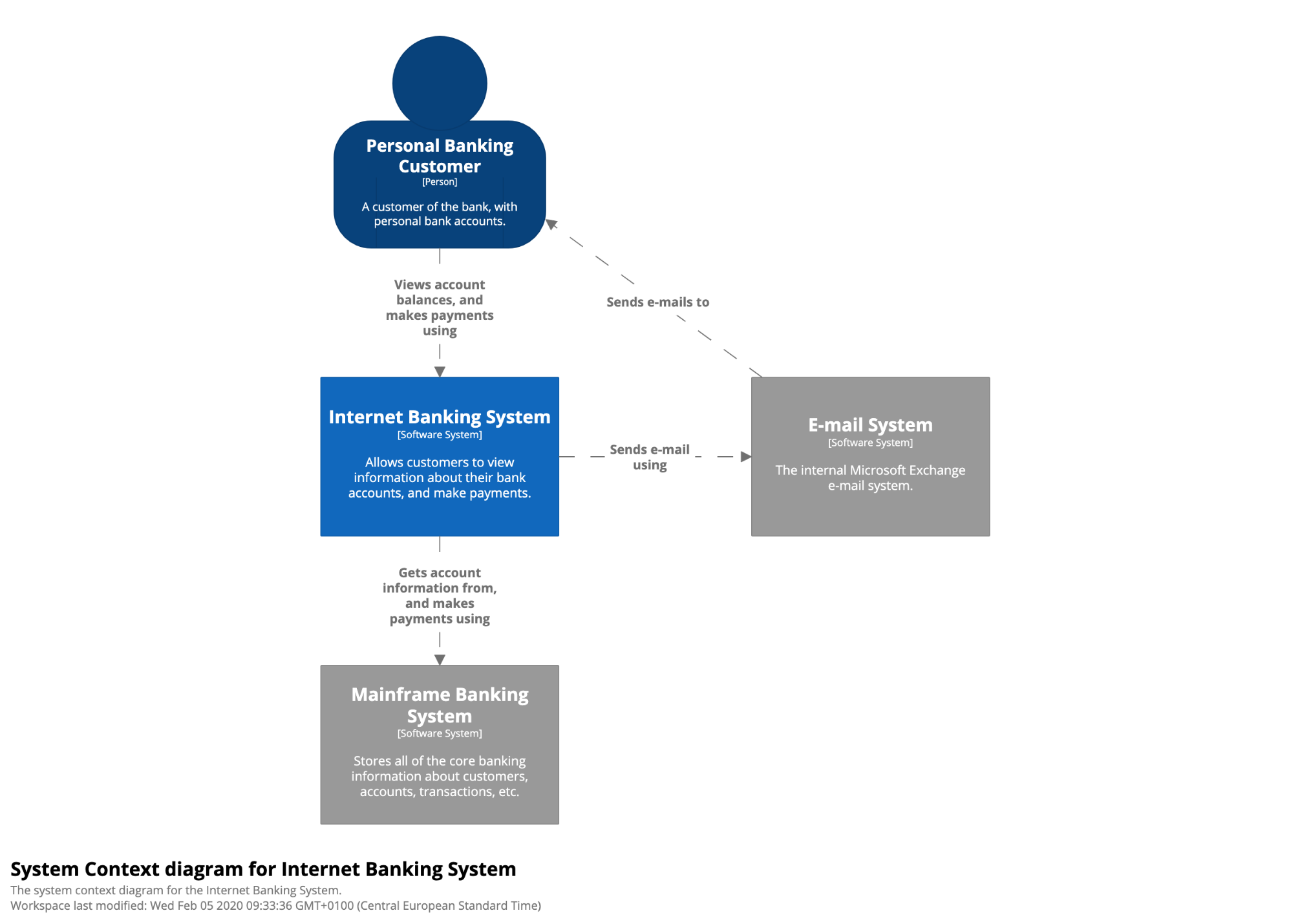
Nivel 4: código;



Es decir, se parte del nivel más alto

Nivel 1: mira nuestra solución como una unidad

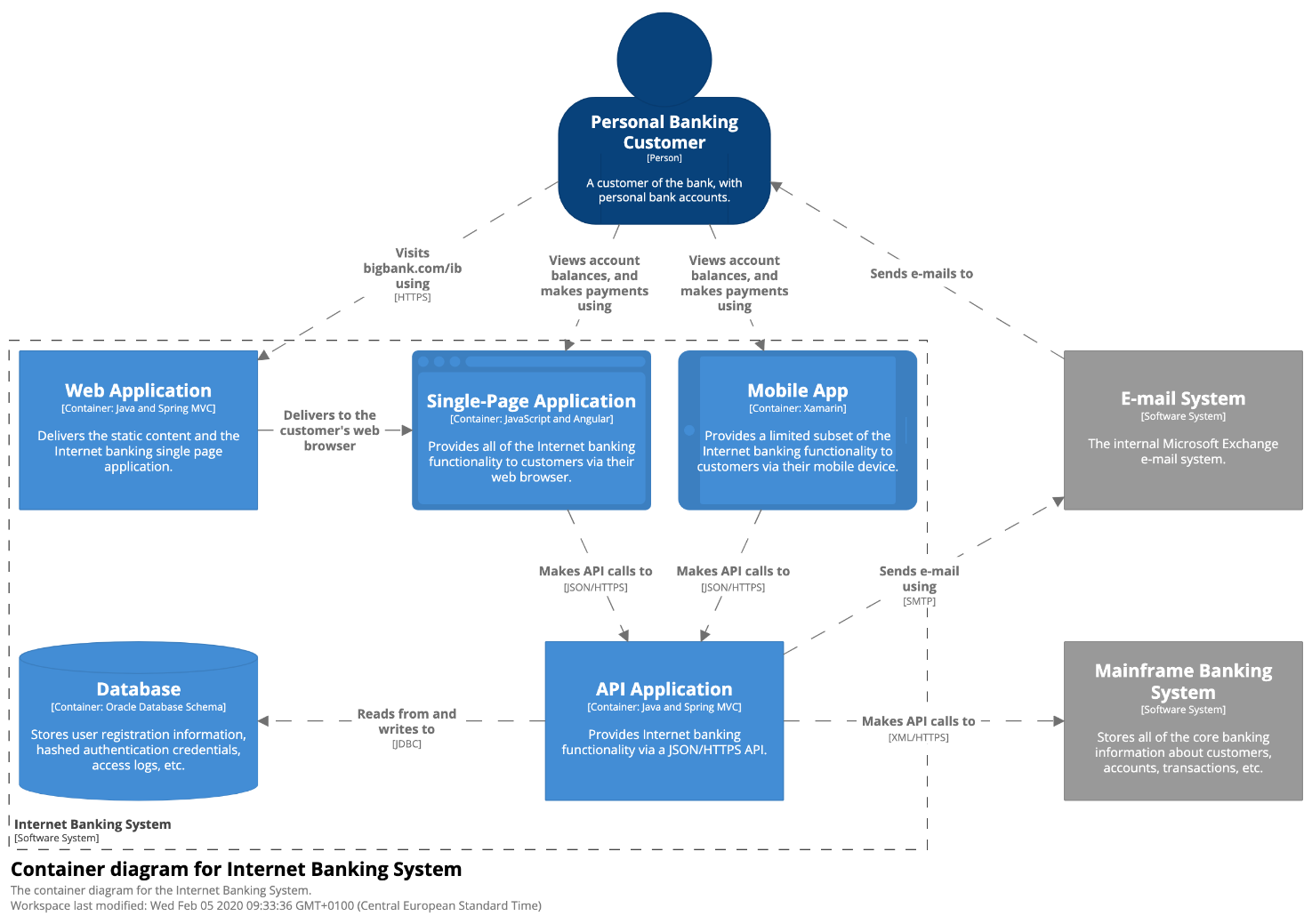
Por ejemplo, acá se muestra un diagrama de contexto una solución que muestra los vecinos en que se interactúa, es decir un pedazo de la arquitectura organizacional, no aspectos internos, sino las interacciones que hay



Nivel 2: un contenedor es algo que desde un puesto de vista de implementación o despliegue es una unidad, como una app web, mobile, un esquema de datos, un file system

Es aquello en que hacemos zoom, hacia debajo de la aplicación, los contenedores que se han definido, es decir de las partes que componen

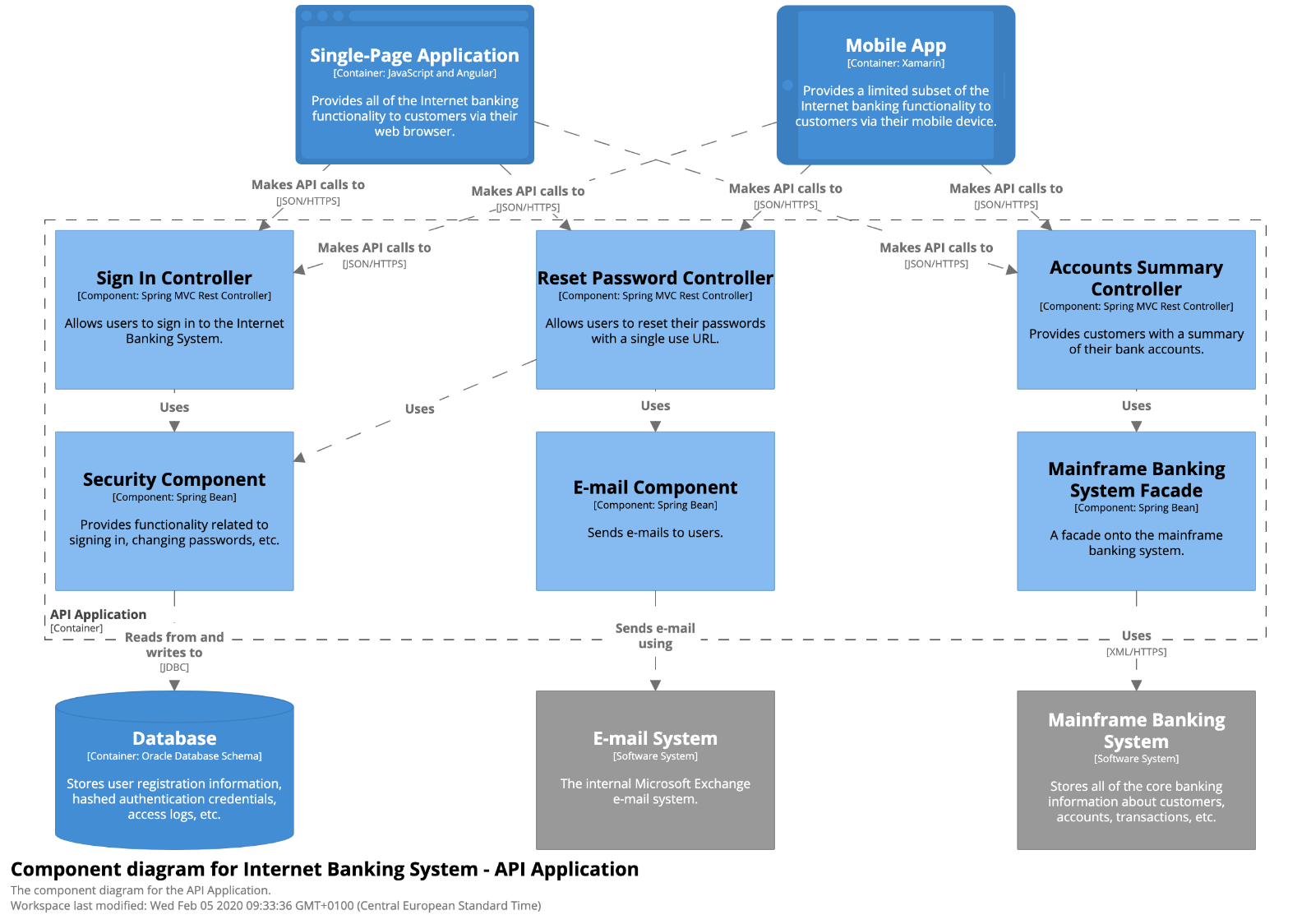
Acá se ve que parte de las aplicaciones se ve afectado y así poder identificar esa parte



## Nivel 3 o de los componentes

En el primer diagrama teníamos aplicaciones, luego nivel 2 es el zoom formado por contenedor

Luego en nivel 3 se ve a través de diagrama de componentes

Ejemplo de dos contenedores del diagrama anterior: 

Como los componentes son compartidos tiene sentido mostrarlo juntos

## Y el nivel 4 o cuando llegamos finalmente el código:

Este paso es mencionado por el hecho de que estamos siendo formales. No olvidemos que los diagramas son herramientas de comunicación, los diagramas no son un fin, es raro ver un diagrama como un elemento a únicamente a ser usado

Los diagramas como herramientas de comunicación, hay que ser selectivo, hay que hacer diagramas que agregue algo, algo importante que refleje que no este perdiendo nada, en el diagrama debe reflejar como usar, pensar el objetivo de ese uso, entonces selectividad a la hora de usar los diagramas.

La arquitectura solida resuelve muchos problemas que tenemos, pero hay una que no resuelve que es vivir en un vecindario

Consultas:

* la arquitectura de una aplicación no resuelve el problema de vivir en un vecindario, ahí están negociaciones, la repartición de recursos, eso repercute la arquitectura de la aplicación, en muchos casos queremos agregar funcionalidad, pero aceptación de otras soluciones que queremos controlar

# ANEXO:

# ¿Qué es la interoperabilidad?

Interoperabilidad es la capacidad que tienen los sistemas y/o equipos no sólo de intercambiar información si no de interpretarla y procesarla en un formato amigable a su usuario.

Nivel de Operabilidad



**¿Cómo se implementa?**

La implementación práctica de la interoperabilidad se basa en el hecho fundamental de compartir interfaces standard, es decir, “Hacer que los sistemas se comuniquen entre sí a través de un lenguaje común”, esto no es más que la implementación de un “**Protocolo**” de comunicación.

Hablar de interoperabilidad refiere a cómo facilitar el intercambio de información entre entidades o sistemas, a la creación de modelos estandarizados con los que puedan representarse e intercambiarse datos acerca de las estructuras más comúnmente encontradas y repetidas, las cuales posiblemente faciliten que paulatinamente se vaya creando, una nube de información con un mínimo de duplicación de información.

¿Y cómo elegir entre tantos estándares disponibles? ¿Qué formatos, protocolos y lenguajes son los más adecuados? ¿Cómo podemos determinar cuál es un estándar abierto?

Los estándares abiertos deberán tener, como mínimo, las características siguientes:

● Disponibilidad.

● Que los derechos de autor estén disponibles, libres de regalías y condiciones.

● Madurez.

● Internacionalmente aceptados.

● De fácil distribución.

● Con amplio soporte en el mercado.

El sólo hecho de reducir la cantidad de datos redundantes ya por sí sólo lo valdría. Agreguemos a esto que reducirá fuertemente la dependencia de proveedores únicos y de la permanencia en el mercado de productos específicos, fomentando el crecimiento de una verdadera industria de desarrollo de software.

**¿Por qué la interoperabilidad es tan importante?**

Principalmente porque facilita en gran medida la gestión conjunta y ordenada de toda la información.

Además entre sus ventajas se puede destacar:

● La adaptabilidad: los distintos sistemas que absorben la información, se conectan y se encargan de distribuir dicha información de forma automática y flexible.

● Cohesión de datos garantizada: con la interoperabilidad, la información es gestionada de forma eficaz y controlada por todas las partes.

● Aumento de la productividad: de la anterior ventaja se puede desprender la idea de que gracias a esta herramienta, se puede asociar toda la información disponible a la cadena de valor. De tal forma, se puede trabajar y operar de forma concordante, a lo largo de todo el proceso productivo, asegurando que la información está disponible y es accesible para todas las partes, sistemas y personas.

**Fuentes:**

[**https://www.tcass.com/menu-notas-de-interes/62-salud/204-que-es-la-interoperabilidad**](https://www.tcass.com/menu-notas-de-interes/62-salud/204-que-es-la-interoperabilidad)

[**https://dej.rae.es/lema/interoperabilidad**](https://dej.rae.es/lema/interoperabilidad)

[**http://interoperability-definition.info/es/**](http://interoperability-definition.info/es/)

[**https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/432/1/M-CD4336.pdf**](https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/432/1/M-CD4336.pdf)

[**https://sg.com.mx/revista/33/programar-es-un-estilo-vida-interoperabilidad**](https://sg.com.mx/revista/33/programar-es-un-estilo-vida-interoperabilidad)

[**https://nexusintegra.io/es/blog/que-es-la-interoperabilidad-y-como-puede-lograrla-mi-em**](https://nexusintegra.io/es/blog/que-es-la-interoperabilidad-y-como-puede-lograrla-mi-empresa/)

[**presa/**](https://nexusintegra.io/es/blog/que-es-la-interoperabilidad-y-como-puede-lograrla-mi-empresa/)

[**https://www.espaciobim.com/interoperabilidad**](https://www.espaciobim.com/interoperabilidad)

[**https://www.chakray.com/es/interoperabilidad-definicion-e-importancia/**](https://www.chakray.com/es/interoperabilidad-definicion-e-importancia/)