



Documentación y diagramación

Silvia Alvarado



Silvia Alvarado

- Staff technical writer en Wizeline
- +10 años de experiencia como technical writer y la industria
- @silviaalvaradom en LinkedIn

Puntos importantes



Identifícate en Zoom utilizando tu nombre y apellido.



Mantén tu micrófono apagado durante el transcurso de la sesión.



Utiliza el chat para hacer tus preguntas durante la sección de Q&A.



Procura enfocar tus preguntas al tema presentado.



Apaga tu cámara en caso de tener problemas con tu conexión.

■ Código de conducta



Sé respetuoso, no hay malas preguntas o ideas.



Sé cordial y paciente.



Sé cuidadoso con tus palabras.

Objetivo

Al final de esta sesión podrás:

- Entender la importancia de la documentación.
- Escribir documentación técnica concisa, clara y centrada en el usuario.
- Diseñar diagramas de arquitectura de software.

Tabla de contenido

Definición de documentación técnica

Y las consecuencias de las malas prácticas de la documentación



Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas



Documentación de arquitecturas de software

Técnicas y tips



Creación de diagramas de arquitectura de software

Herramientas y tips



Versionamiento de documentos



■ **Pregunta rompehielos**

¿Cuál es tu opinión sobre los documentos y la documentación técnica?



■ **Definición de documentación técnica**

Y las consecuencias de las malas prácticas de la documentación

“La documentación técnica es la disciplina que **transforma la información técnica** en documentos útiles acerca de **productos, procesos o servicios** enfocados a distintas audiencias con el fin de **instruirlos, informarlos o guiarlos** a sus objetivos.”

¿Qué es la documentación técnica?

¿Para qué sirven los documentos?

- Son herramientas esenciales para garantizar la comunicación y la comprensión entre los diferentes equipos de una organización, así como para facilitar la toma de decisiones.
 - Guías de inducción
- Son registros históricos que nos proporcionan la información detallada y cronológica de eventos, decisiones, diseños, implementaciones y cambios a lo largo del tiempo.
 - Historial de versiones
 - Documentos de arquitectura
 - Documentación de requerimientos
- Herramientas de marketing que permiten generar clientes potenciales al abordar problemas del sector o producto incluyendo palabras clave específicas para subir en las clasificaciones de búsqueda.
 - Guías de usuario final
 - Guías de resolución de problemas

Consecuencias de las malas prácticas de documentación en las empresas

Dependencia del conocimiento del personal

Inducción de empleados sin la documentación adecuada

Falta de documentación de procesos críticos

Documentación mal estructurada

Falta de buenas prácticas en la documentación

Documentación desactualizada

Tabla de contenido

Definición de documentación técnica

Y las consecuencias de las malas prácticas de la documentación



Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas



Documentación de arquitecturas de software

Técnicas y tips



Creación de diagramas de arquitectura de software

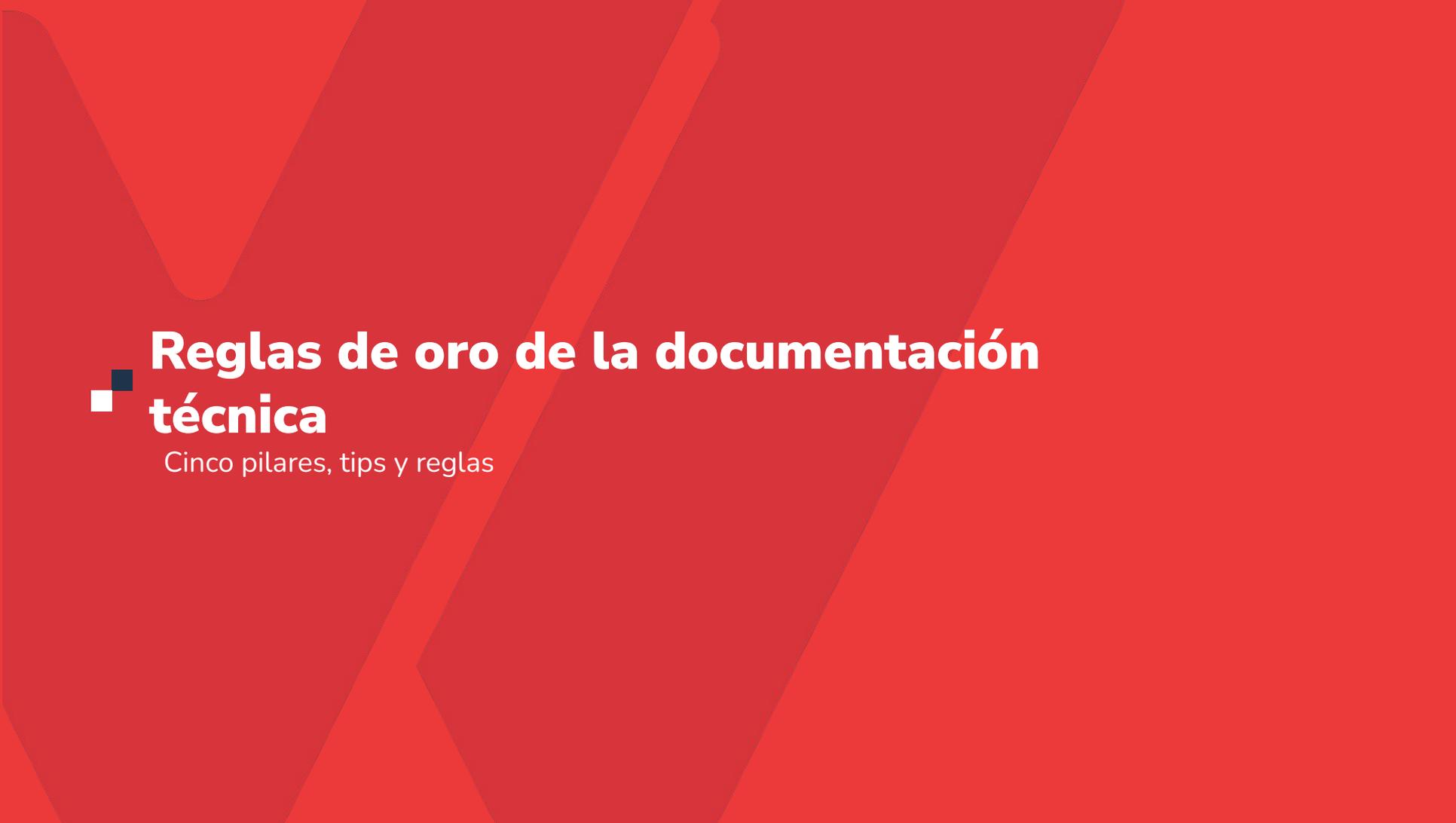
Herramientas y tips



Versionamiento de documentos

Uso e importancia

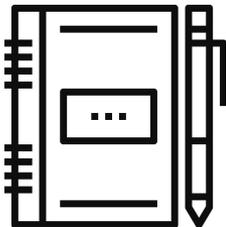




■ ■ Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas

Cinco pilares



- Conoce a tu **audiencia**
- **Investiga** el tema
- Se **conciso**
- **Organiza** la información de forma lógica
- Usa gráficas, imágenes y tablas **moderadamente**



“Si tuviera más tiempo, te habría escrito una carta más corta”

Blaise Pascal



Aplicación de seguro de vehículos

Reporta tu siniestro

La aplicación de seguros cuenta con un UI fácil de usar y un poderoso backend que soporta miles de consultas por minuto.

1. Si no tienes cuenta, no podrás realizar el reporte.
2. Si tienes cuenta, revisa la sección de Generación del Reporte de Siniestro.

Vale la pena mencionar que la aplicación es una solución alojada en la nube de AWS.



Servicios del API

Para cancelar una solicitud de pago debes usar el API que la empresa ofrece de manera pública para que puedas usarlo dentro de tus productos. La API te permite acceder a los ciertos recursos dentro de nuestro servicio, así como agregar datos, borrarlos o editarlos. Por ejemplo, para cancelar una solicitud, puedes usar el endpoint DELETE `https://api-gw.payclip.com/paymentrequest/code/{payment_request_code}`. Recuerda que necesitas tu token de autenticación para realizar las llamadas a la API. El único dato que necesitas es `payment_request_code`.



Uso de negritas, cursivas y fuentes monoespaciadas

Considera lo siguiente:

- Usa **negritas** para resaltar:
 - Nombres de los elementos en la interfaz de usuario.
Haz clic en el botón **Crear** para comenzar el proceso.
Ingresa la dirección IP de tu sistema en el campo **Dirección IP**.
 - Entradas del usuario.
Ingresa **10.1.1.1** en el campo **Dirección IP**.
- Usa *cursivas* o itálicas para lo siguiente:
 - Dar énfasis.
No debes entrar al edificio sin identificación.
 - Resaltar palabras en otro idioma.
Crea un pull request para someter tus cambios a revisión.
- Usa fuentes monoespaciadas para resaltar:
 - Comandos o fragmentos de código.
Ejecuta el comando `npm start` para iniciar el servidor de desarrollo local.
 - Nombres de archivos o carpetas.
Lee el archivo `README.md` antes de proceder.

Siglas, acrónimos y abreviaturas

- Usa las siglas y acrónimos según lo familiarizada que esté la audiencia con ellas:
 - **Siglas o acrónimos.** Escribe primero las siglas o acrónimos de uso común y después desglosa entre paréntesis las palabras que las conforman.
Los usuarios de la aplicación pueden compartir su URL (localizador uniforme de recursos, por sus siglas en inglés) personal.
 - **Siglas o acrónimos traducidos.** Si debes ser más preciso con tu traducción, desglosa primero las palabras en español y luego, entre paréntesis, escribe las siglas y cómo se desglosan en el idioma de origen.
Interfaz gráfica de usuario (GUI, graphical user interface).
- Evita usar abreviaturas en documentos formales. Puedes usar en cambio:
 - p. ej., v. gr., v. g. -> por ejemplo
 - c. e. -> correo electrónico
 - cía., comp. -> compañía
 - sig., s. -> siguiente

Uso de viñetas y listas numeradas

Viñetas

- Las viñetas en una lista son útiles cuando el orden de los elementos que la conforman no es importante. Por ejemplo:
 - Wizeline Academy realiza cursos sobre:
 - UX
 - Programación
 - Technical writing

Listas numeradas

- Las listas numeradas son útiles cuando el orden de los elementos que la conforman es importante. Por ejemplo:
 - Completa las tareas requeridas de la siguiente manera:
 - i. Configura la cuenta **MyCompany Okta**.
 - ii. Configura la VPN y el certificado de **MyCompany** en tu computadora local.
 1. Firma y envía el formulario de acceso **Vendor Remote Access a MyCompany**.
 2. Espera a que MyCompany apruebe el acceso.
 3. Comunícate con el **Service Desk de MyCompany** para una sesión de configuración mediante acceso remoto.

Nota: Personas con las que deberás comunicarte en MyCompany respecto a la configuración:

 - John Doe, Director de TI.
 - Jane Smith, Service Desk Lead.
 - iii. Conéctate a la VPN de **MyCompany**.
 - iv. Confirma que tienes acceso al repositorio del proyecto en Repo / MyCompany Jira.

Aplicación de seguro de vehículos

Reporta tu siniestro

La aplicación de seguros cuenta con un UI fácil de usar y un poderoso backend que soporta miles de consultas por minuto.

1. Si no tienes cuenta, no podrás realizar el reporte.
2. Si tienes cuenta, revisa la sección de Generación del Reporte de Siniestro.

Vale la pena mencionar que la aplicación es una solución alojada en la nube de AWS.



Aplicación de seguro de vehículos

Reporta tu siniestro

Cuando sufra un siniestro vehicular como ser cualquiera de los siguientes:

- **Accidentes viales.** Se refiere a cuando choque con otro vehículo...
- **Daños a terceros.** Que va desde atropellamiento a peatones hasta daños causados a otro auto.
- **Robo total.**
- ...

Usted puede utilizar la aplicación móvil para hacer el registro de su siniestro al momento del mismo.

...



Servicios del API

Para cancelar una solicitud de pago debes usar el API que la empresa ofrece de manera pública para que puedas usarlo dentro de tus productos. La API te permite acceder a los ciertos recursos dentro de nuestro servicio, así como agregar datos, borrarlos o editarlos. Por ejemplo, para cancelar una solicitud, puedes usar el endpoint DELETE `https://api-gw.payclip.com/paymentrequest/code/{payment_request_code}`. Recuerda que necesitas tu token de autenticación para realizar las llamadas a la API. El único dato que necesitas es `payment_request_code`.



Servicios del API

Para cancelar una solicitud de pago debes usar el API que la empresa ofrece de manera pública para que puedas usarlo dentro de tus productos.

Usa el endpoint:

DELETE

https://api.empresa.com/paymentrequest/code/{payment_request_code}

Authentication: token

Parametro:

- **payment_request_code** (String): Código de registro. Lo puedes obtener ejecutando...



Tabla de contenido

Definición de documentación técnica

Y las consecuencias de las malas prácticas de la documentación



Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas



Documentación de arquitecturas de software

Técnicas y tips



Creación de diagramas de arquitectura de software

Herramientas y tips



Versionamiento de documentos

Uso e importancia



■ **Documentación de arquitecturas de software**

Técnicas y tips

Resumen

Documentar arquitecturas de software es como hacer zoom a un mapa de Google Maps.



¿Qué es una arquitectura de software?

La arquitectura de software se refiere a las partes de un sistema y la forma en que se combinan entre ellas.

La arquitectura de software se basa en **patrones, modelos y abstracciones teóricas**, que nos permita entender cómo encajan cada una de las piezas de nuestro producto o servicio.

La arquitectura de software proporciona una visión de alto nivel de cómo funcionará el sistema en su conjunto y cómo se dividirá en componentes más pequeños que trabajan juntos para lograr los objetivos de software.



**“El código documenta
el sistema, no
necesitamos más
documentación”**



¿Por qué documentar una arquitectura de software?

Porque hay cosas que el código no especifica,
como:

- ¿Por qué está construido el sistema?
- ¿Cómo funciona el sistema en general?
- ¿Cuáles son los objetivos del sistema?
- ¿Quién necesita entender el sistema?

Documentación de arquitectura de software

Objetivo

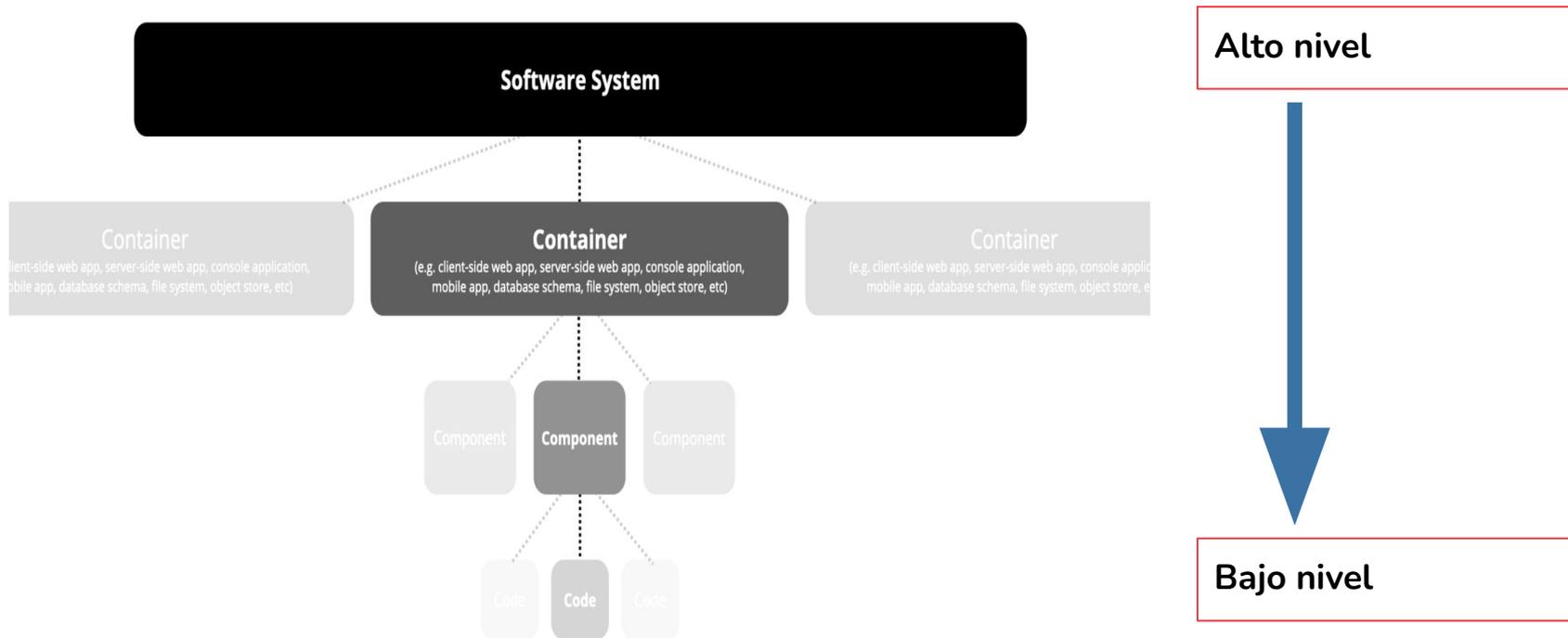


- Crea una comprensión común de la solución
- Ayuda a comprender desde diferentes perspectivas
- Apoya el trabajo del equipo
- Facilita la transferencia de conocimientos
- Ayuda a las **evaluaciones de seguridad**

Tips para documentar arquitecturas de software

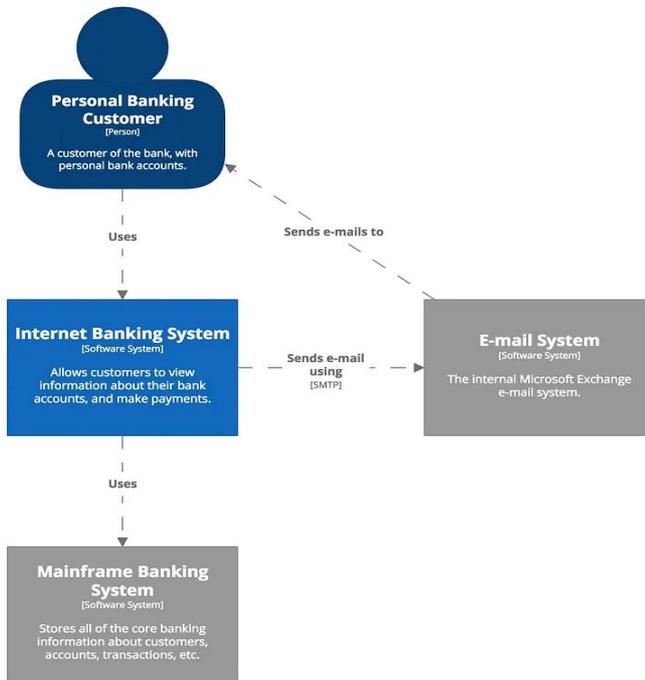
- Define la audiencia del documento.
- Entiende cómo funciona el patrón de arquitectura.
- Aplica la técnica C4 u otra técnica de visualización de arquitecturas.
- Se consistente con los términos técnicos usados en el documento.

C4: Contexto, contenedores, componentes y código



Nota: Usamos C4 como generalmente, pero para sistemas más complejos, usar ERD, diagramas de secuencias, entre otros.

Visualización de la arquitectura de software: Contexto

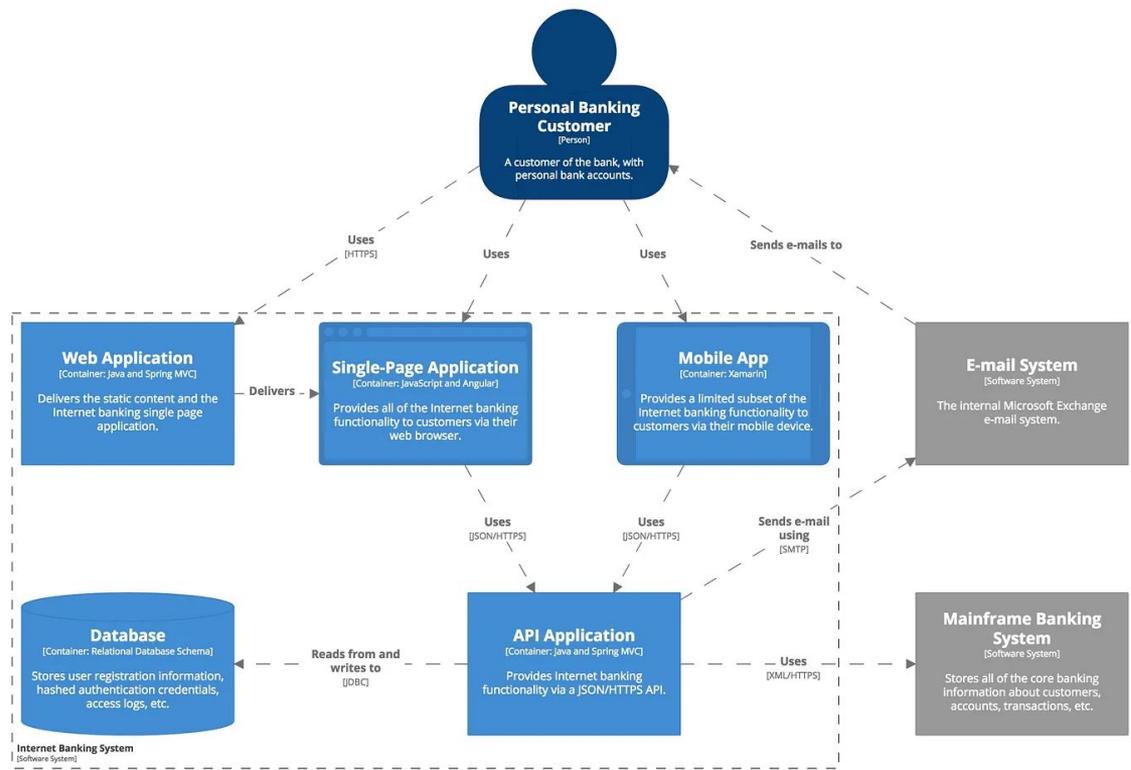


Muestra el sistema de software que está construyendo y cómo encaja en el mundo en términos de las personas que lo utilizan y los otros sistemas de software con los que interactúa.

System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.
Last modified: Wednesday 02 May 2018 13:46 UTC

Visualización de la arquitectura de software: Contenedor

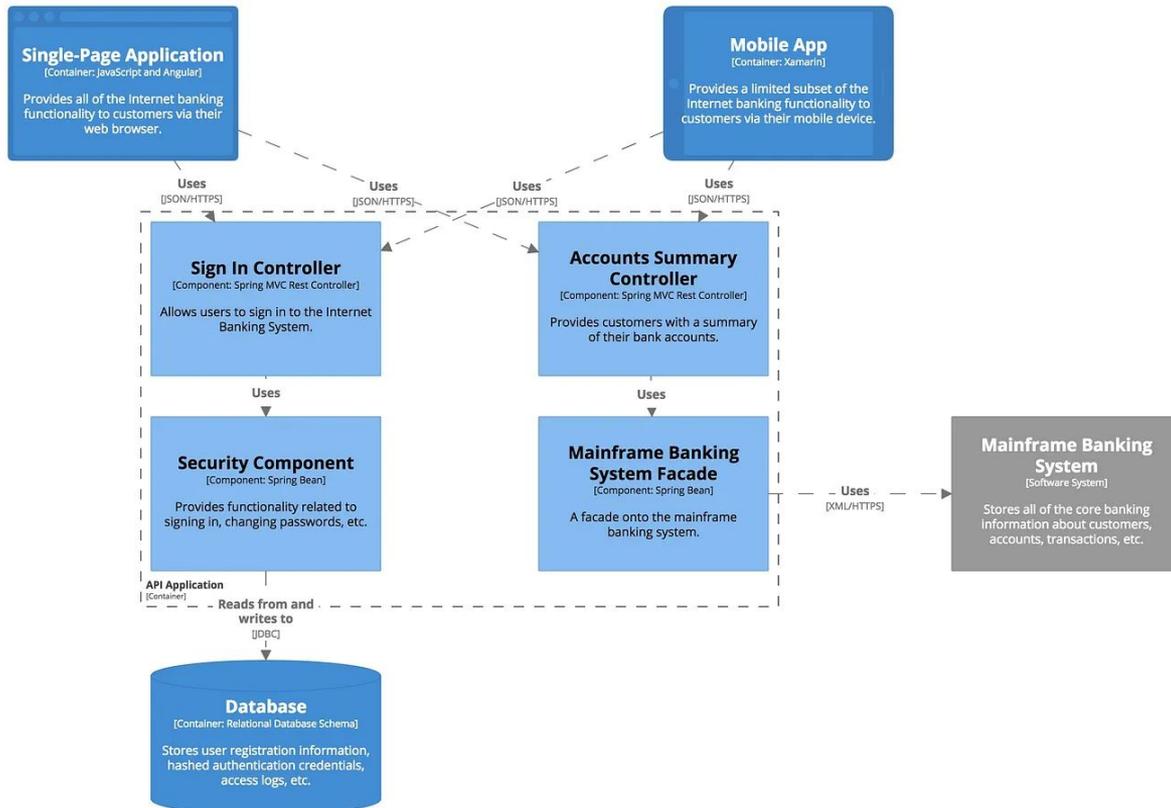


Container diagram for Internet Banking System

The container diagram for the Internet Banking System.
Last modified: Wednesday 02 May 2016 13:46 UTC

Muestra los contenedores (aplicaciones, almacenamiento de datos, microservicios, etc.) que componen este sistema de software.

Visualización de la arquitectura de software: Componentes

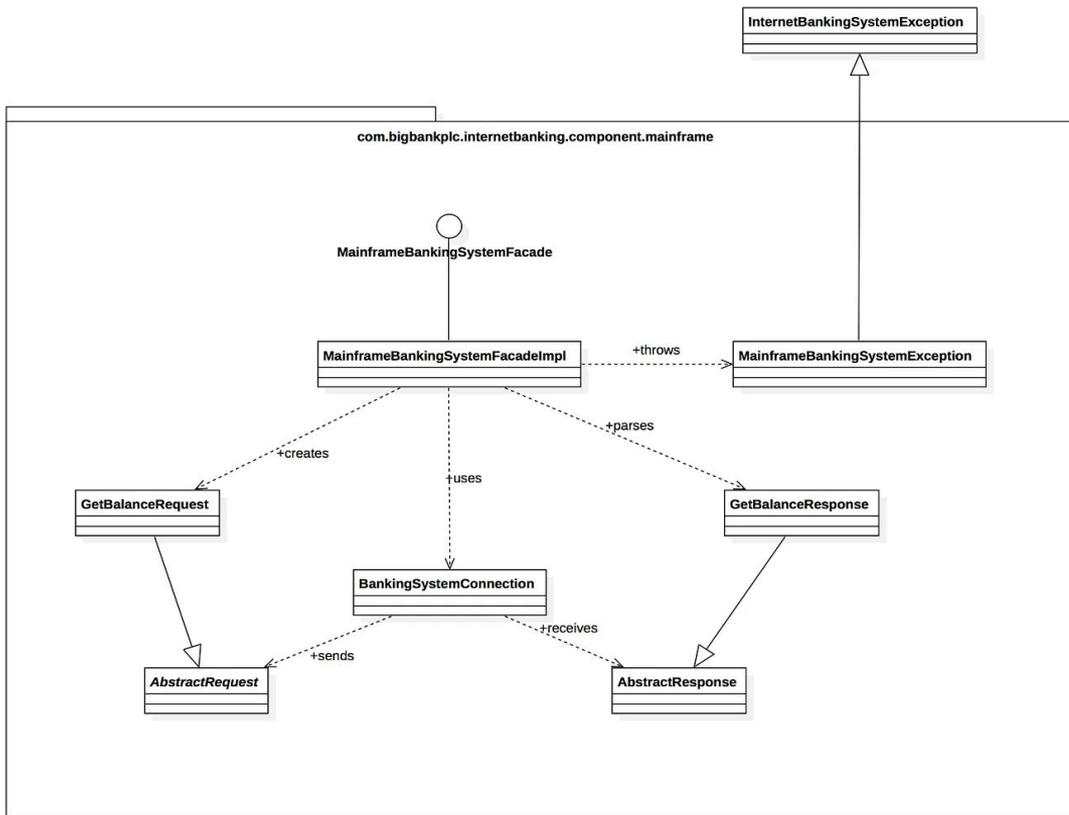


Expande un contenedor individual para mostrar los componentes que contiene. Estos componentes deben asignarse a abstracciones reales (por ejemplo, una agrupación de códigos) en función de su código.

Component diagram for Internet Banking System - API Application

The component diagram for the API Application.
Last modified: Wednesday 02 May 2018 13:46 UTC

Visualización de la arquitectura de software: Código



Muestra cómo se implementa este componente.

Documentación de arquitectura de software

Imaginemos que la siguiente es la arquitectura a alto nivel de YouTube.

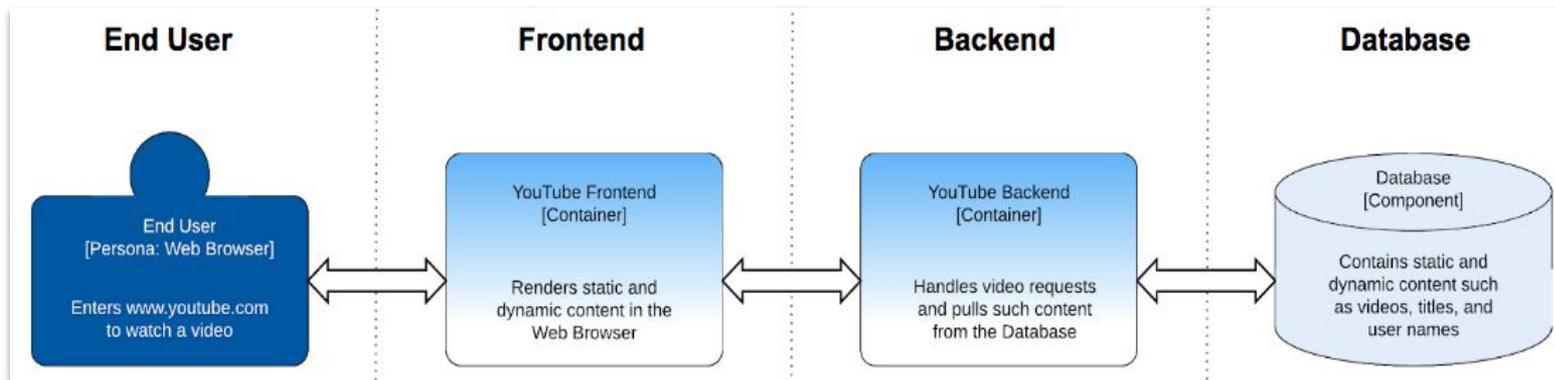


Diagrama de alto nivel

Documentación de arquitectura de software

Introducción

Diagrama

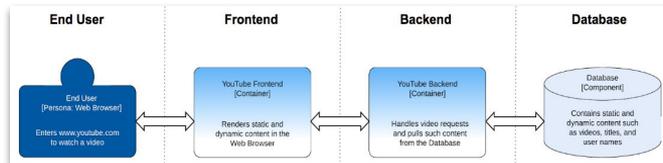
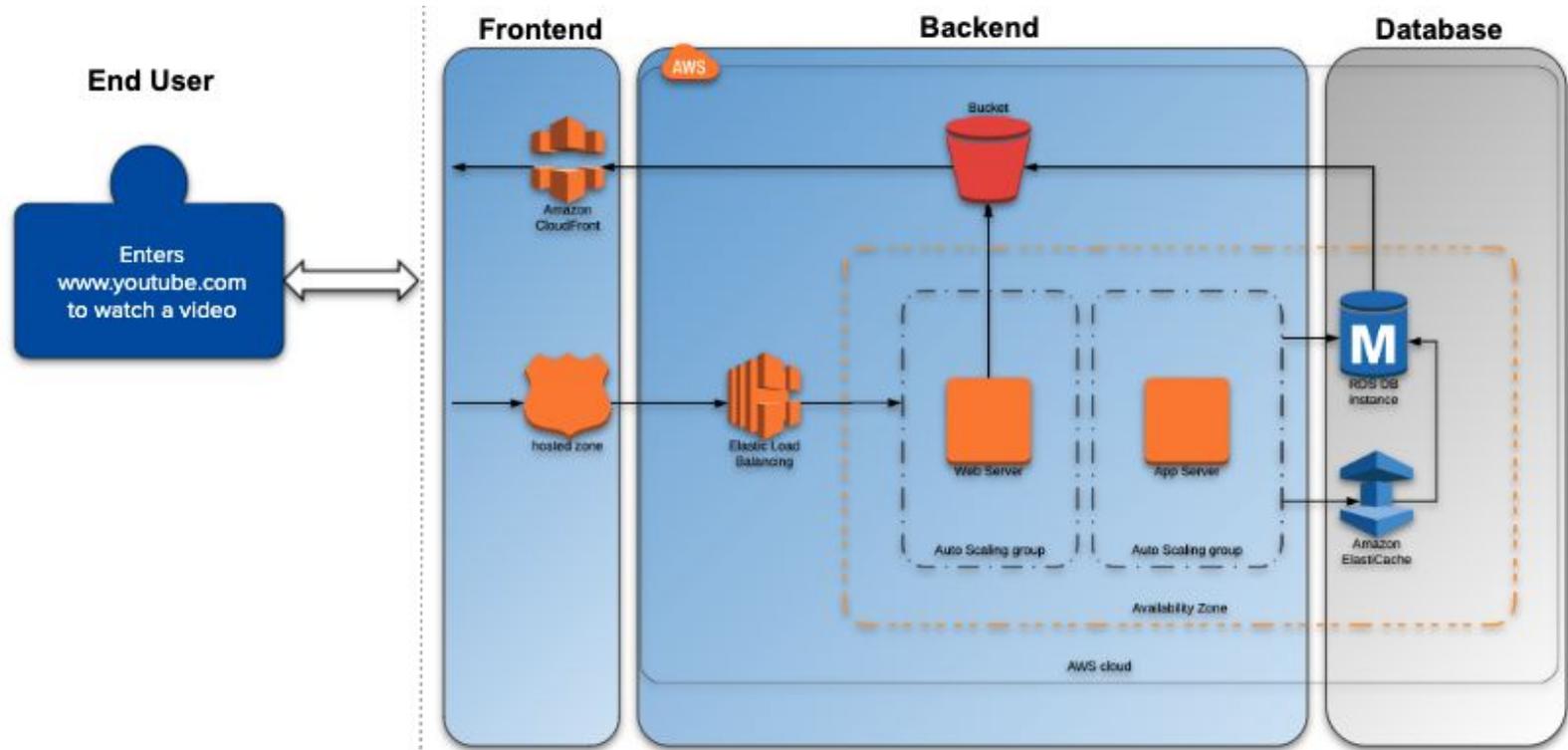


Diagrama de alto nivel

- **Frontend**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Backend**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Database**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)

Documentación de arquitectura de software



Documentación de arquitectura de software

Introducción Diagrama

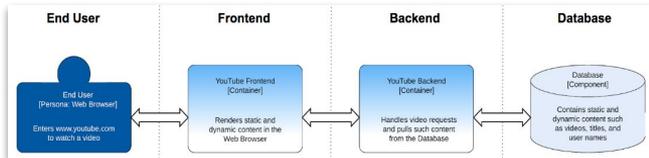


Diagrama general

- **Frontend**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Backend**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Database**
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)

Introducción Diagrama

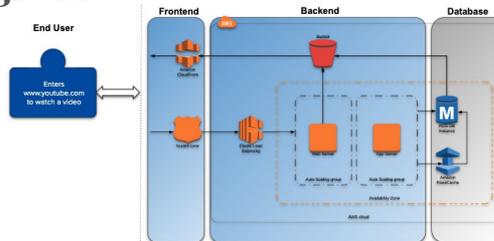


Diagrama detallado

- **Frontend**
 - Elements (Name, Definition)
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Backend**
 - Elements (Name, Definition)
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)
- **Database**
 - Elements (Name, Definition)
 - Propiedades (Responsabilidades)
 - Relaciones (Comunicación)



Documentación de arquitectura de software

Introducción

YouTube es una plataforma que permite ver y subir videos a creadores y consumidores de contenido.

Detalles de la plataforma

La plataforma de Youtube está alojada en la nube e incluye los siguientes componentes:

- **End User (usuario final)**: Consume contenido a través de la interfaz de YouTube.
- **Frontend (interfaz)**: Despliega el contenido solicitado por el usuario final.
- **Backend**: Maneja peticiones desde el frontend y obtiene contenido desde la base de datos.
- **Database (base de datos)**: Alcena el contenido multimedia.

La estructura de YouTube facilita el flujo de datos desde la base de datos hasta el usuario final y viceversa.



Documentación de arquitectura de software

End user

El usuario final (end user) accede e interactúa con la interfaz (*frontend*) dentro de un navegador web. El usuario final hace peticiones de contenido y recibe el contenido solicitado en la interfaz.

Frontend

La interfaz (*frontend*) despliega contenido estatico y dinamico. El *frontend* facilita la experiencia del usuario a través de su aplicación responsiva. Este componente le permite al usuario final solicitar contenido desde el backend y la base de datos.

Backend

El backend maneja el contenido de todas las peticiones enviadas desde el frontend. Localiza y obtiene contenido estatico y dinamico desde la base de datos. Este componente conecta el frontend a la base de datos.

Tabla de contenido

Definición de documentación técnica

Y las consecuencias de las malas prácticas de la documentación



Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas



Documentación de arquitecturas de software

Técnicas y tips



Creación de diagramas de arquitectura de software

Herramientas y tips



Versionamiento de documentos

Uso e importancia



■ Creación de diagramas de arquitectura de software

Herramientas y tips

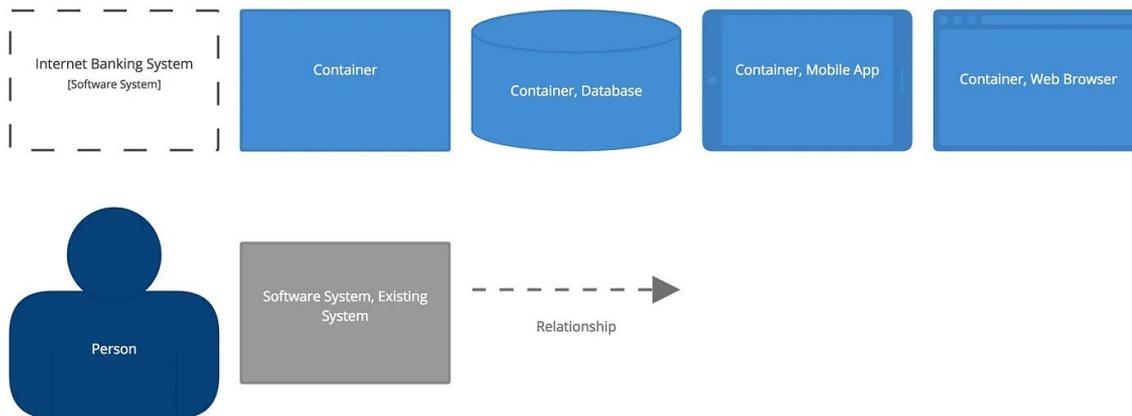
Herramientas para el diseño de arquitecturas

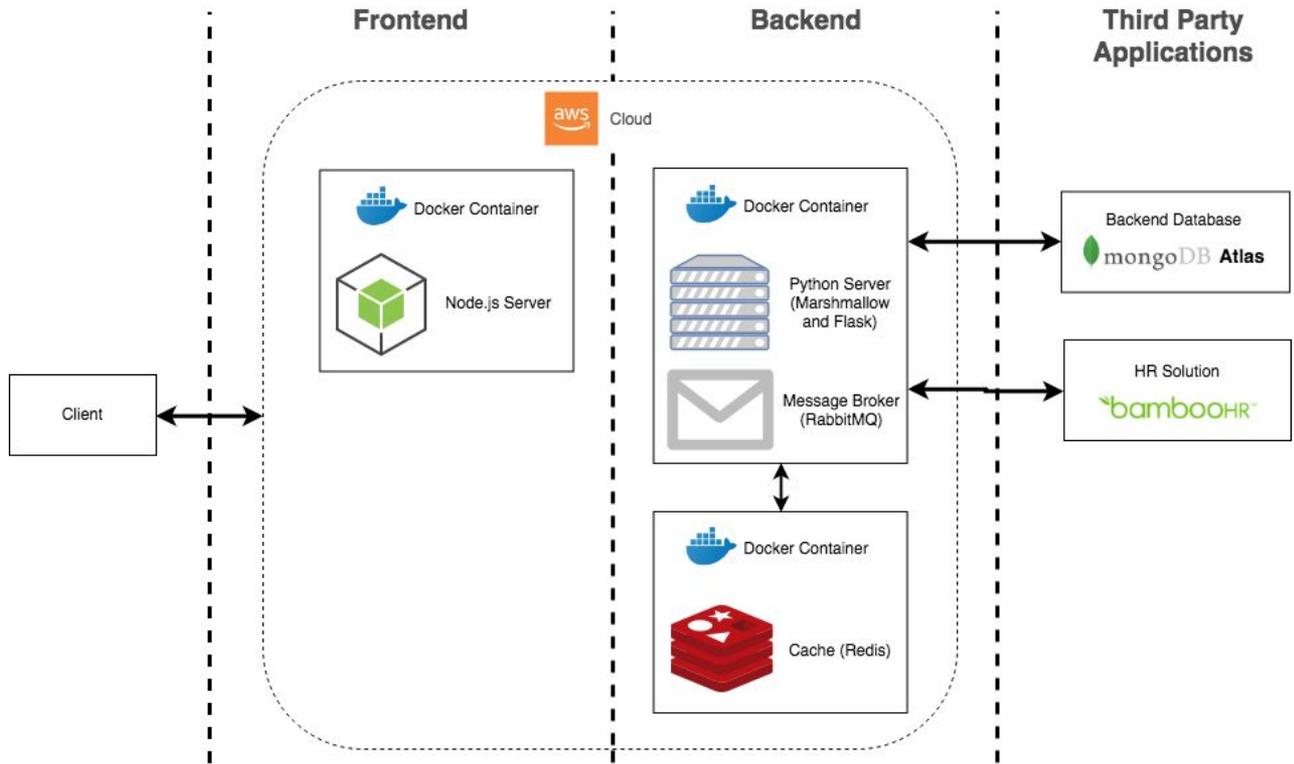


[Las 8 mejores herramientas de diagramación](#)

■ Reglas de oro para diseñar diagramas de arquitectura

- Identifica el propósito del diagrama.
- Entiende el flujo de datos en la arquitectura de software.
- Utiliza una notación estándar.
- Incluye información relevante dentro de las interacciones de los componentes.
- Divide los componentes de software siguiendo el modelo C4.
- Utiliza los iconos correspondientes a los servicios.





Lucidchart

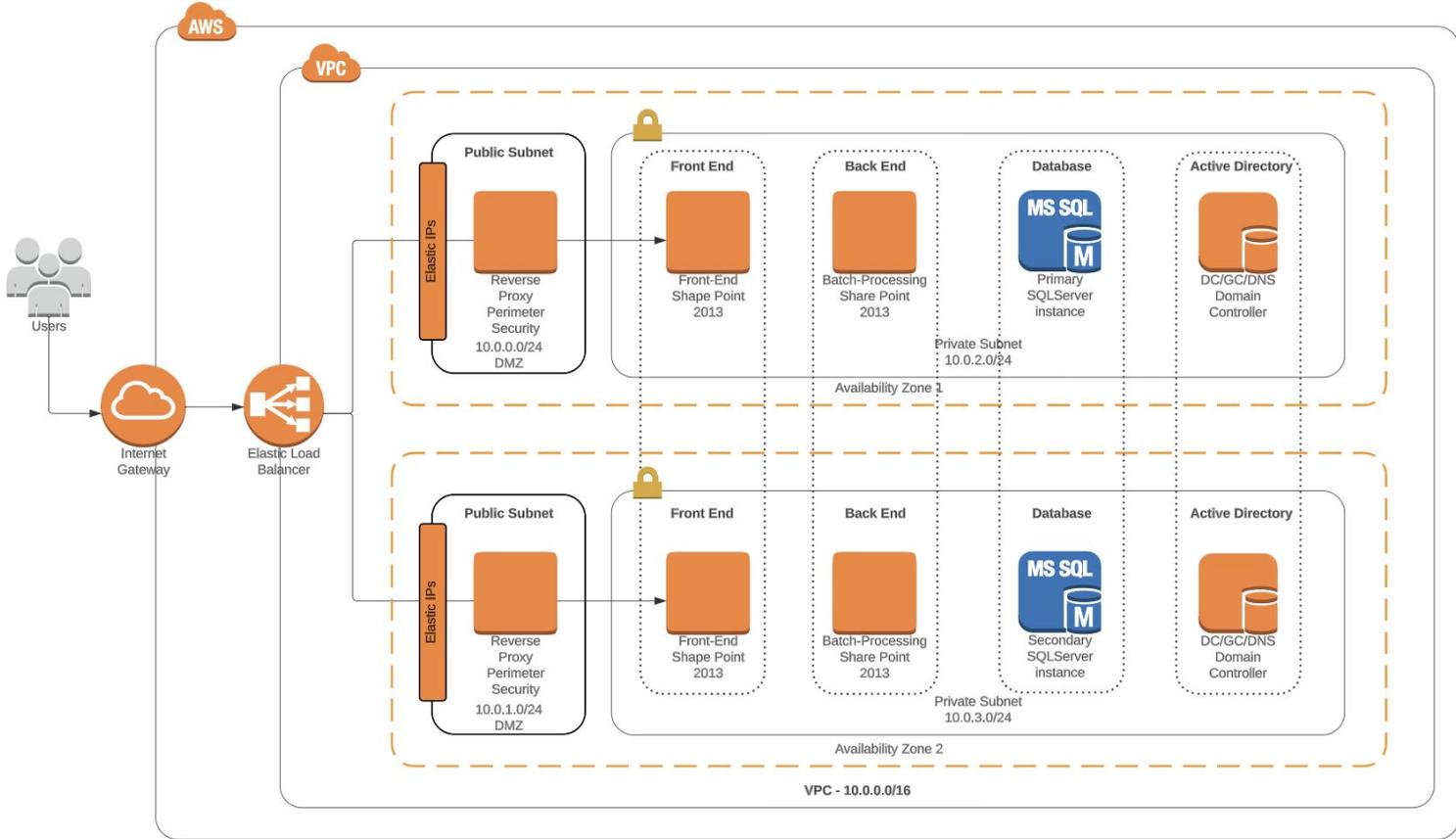


Tabla de contenido

Definición de documentación técnica

Y las consecuencias de las malas prácticas de la documentación



Reglas de oro de la documentación técnica

Cinco pilares, tips y reglas



Documentación de arquitecturas de software

Técnicas y tips



Creación de diagramas de arquitectura de software

Herramientas y tips



Versionamiento de documentos

Uso e importancia





Versionamiento de documentos

Uso e importancia

Sobre el versionamiento...

Existen varias herramientas de control de versiones (VCS, por sus siglas en inglés) que son utilizadas para gestionar y controlar las versiones de software. Algunas de las más populares son:

- **Git:** Es distribuido y permite el control de versiones de manera eficiente. GitHub, GitLab y Bitbucket son plataformas populares que utilizan Git para alojar repositorios y ofrecen funciones adicionales, como seguimiento de problemas y colaboración.
- **Subversion (SVN):** SVN es un sistema de control de versiones centralizado. A diferencia de Git, SVN sigue un modelo centralizado donde un servidor central almacena el repositorio y los usuarios realizan operaciones directamente en él.
- **Mercurial:** Similar a Git, Mercurial es otro sistema de control de versiones distribuido.



Importancia del versionamiento

El control de versiones es fundamental en el desarrollo de software y en muchos otros contextos donde se gestionan archivos y documentos.

- Historial de cambios
- Colaboración
- Reversión a versiones anteriores
- Implementación continua
- Auditoría



Changelogs

Documento que proporciona un registro detallado de todas las modificaciones realizadas en un software o proyecto a lo largo del tiempo.

```
# Changelog
```

```
## [Versión 1.2.0] - 2023-11-22
```

```
### Cambios
```

- ****Nueva Característica:**** Se ha agregado la capacidad de importar y exportar datos.
- ****Mejora de Rendimiento:**** Optimización del algoritmo de búsqueda para una respuesta más rápida.
- ****Corrección de Errores:**** Se soluciona un problema que causaba cierres inesperados al realizar ciertas acciones.

```
## [Versión 1.1.0] - 2023-10-15
```

```
### Cambios
```

- ****Mejora de Interfaz de Usuario:**** Rediseño de la interfaz de usuario para mejorar la

Cómo versionar los documentos manualmente

Versión	Registro de cambios	Fecha	Revisor

- Se debe establecer una estrategia de nomenclatura como ser:
 - Documento_v1.0.docx
 - Documento_v1.1.docx
 - Documento_v2.0.docx
 - Documento_2023-11-22_v1.0.docx
 - Documento_Especificaciones_v1.2.docx
- Agregar la fecha de la versión si la frecuencia de las actualizaciones es alta o si se necesita una referencia adicional.
-

Resumen general



- ¿Qué es lo más útil de la lección?
- ¿En qué documentos van a aplicar estos tips?

¿Preguntas?





Gracias