 **Universidad de Guayaquil**

**Facultad de ingeniería industrial**

**Nombre:**

Ariel Marlon Bonilla Llanos

**Curso**

SIN-S-MA 4-1

**Docente**

Ing. Erick Guerrero

**Tema**

Portafolios Unidad #3

**Periodo Lectivo**

2020

Contenido

[**Justificación** 3](#_Toc51588809)

[**24/08/2020: Componentes y sistemas distribuidos.** 4](#_Toc51588810)

[**31/08/2020: Arquitectura orientada a servicios.** 5](#_Toc51588811)

[**02/09/2020: Arquitecturas de sistemas distribuidos.** 10](#_Toc51588812)

[**07/09/2020: Que es DevOps.** 14](#_Toc51588813)

[**14/09/2020: Gif Lab.** 16](#_Toc51588814)

[**Glosario** 20](#_Toc51588815)

[**Bibliografía** 22](#_Toc51588816)

# **Justificación**

"El interés que gran parte de los alumnos muestra respecto a Internet y todo lo que tiene que ver con esta ha sido clave para proyectar la realización de material didáctico a través de una página web.

El hecho de que a los alumnos se les presente la información en una forma fuera de lo común genera en ellos una gran disposición por conocer el contenido de la misma.

El hipertexto contenido en las páginas web constituye una gran ventaja, pues, gracias a él, el alumno podrá interactuar con el material, con lo cual su labor no se restringirá a la de mero espectador. Además, podrá consultar la información que a él le interese cuantas veces lo crea necesario." (Campos, 2019)

**24/08/2020: Componentes y sistemas distribuidos.**

Los sistemas distribuidos son sistemas cuyos componentes hardware y software, que están en computadoras conectadas en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo preestablecido.

**Componentes y sistemas distribuidos.**

**Características**

**Desventajas**

Compartir recursos (HW, SW, datos).

Necesidad de software mas complejo

Ofrecen una buena relación coste/rendimiento.

Capacidad de crecimiento

Problemas de fiabilidad

Tolerancia a fallos, disponibilidad

Concurrencia

Problemas de seguridad y confidencialidad.

Velocidad

Un sistema distribuido**es un conjunto de equipos independientes que actúan de forma transparente actuando como un único equipo.**

Algunas de las ventajas que aporta un sistema distribuido son:

* Mayor eficacia.
* Mayor tolerancia a fallos: al estar distribuida la información en nodos, en caso de que se caiga un nodo, dicha información va a encontrarse replicada en otros nodos.
* Mayor velocidad y procesamiento distribuido: cuando se realiza una consulta, lo procesamientos se dividen entre todos los nodos que forman el sistema distribuido, en lugar de enviarlos a un único nodo y que el mismo tenga que hacer todo el trabajo.
* Escalabilidad: si, por ejemplo, se necesita más procesamiento o añadir más disco duro, en lugar de que los equipos crezcan de forma vertical añadiendo más almacenamiento, RAM o CPU, se añaden equipos de forma horizontal al clúster o sistema distribuido. (Losada, 2018)

# **31/08/2020: Arquitectura orientada a servicios.**

**Servicios web:** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

**SOAP:** es un [protocolo](https://es.wikipedia.org/wiki/Protocolo_(inform%C3%A1tica)) [estándar](https://es.wikipedia.org/wiki/Norma_(tecnolog%C3%ADa)) que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos [XML](https://es.wikipedia.org/wiki/XML).

**RESTful:** un servicio Web RESTful es aquel servicio web que está basado en la **arquitectura REST**. Los servicios Web RESTful se basan en **recursos**. Un recurso es una entidad, la cual se almacena principalmente en un servidor y el cliente solicita el recurso utilizando servicios **Web RESTful**.

**Servicios web**

**Definición**

**Características**

Interoperabilidad

Extensibilidad

Es un sistema software diseñado para soportar interacciones maquina a maquina a través de la red (W3C World Wide Web Consortium)

Pueden acoplarse para conseguir operaciones complejas

Información procesable XML



**Tipos de servicios web**

Servicios web ligeros

Servicios Web Grandes

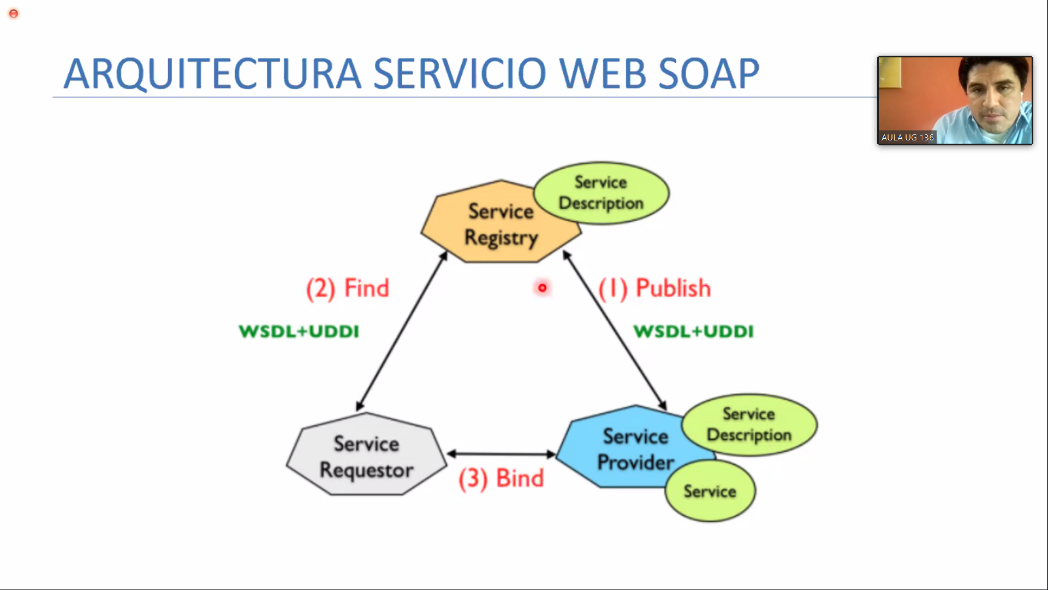
Servicios Web SOAP

Servicios Web RESTful.

**Servicios Web SOAP**

Se caracterizan:

Arquitectura Servicios Web SOAP



Por utilizar mensajes XML que siguen el estándar SOAP

Tecnologías básicas servicios web SOAP

Tecnologías básicas servicios web SOAP

Posee dos tipos de mensajes:

Transporte de servicios

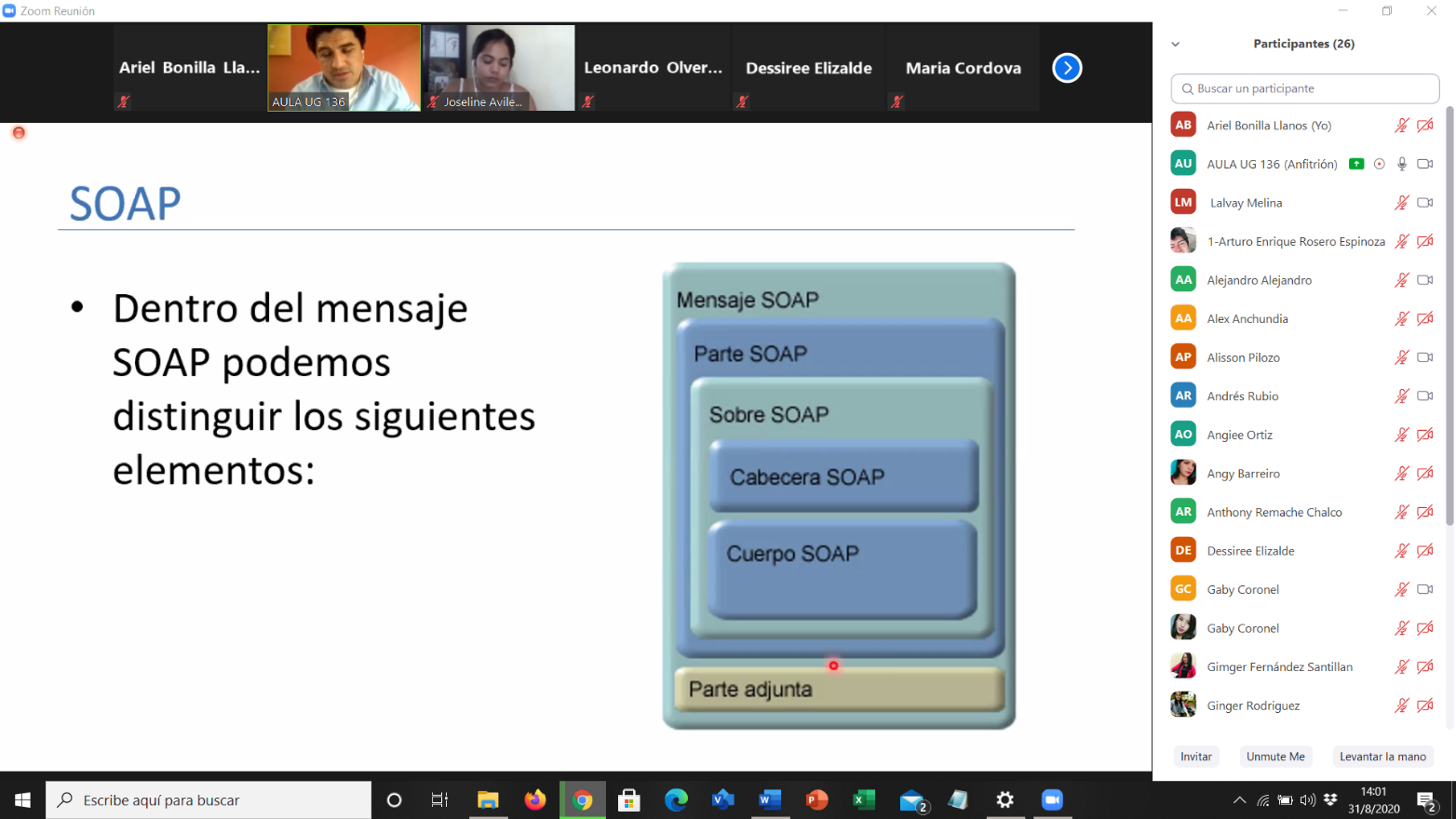
Mensajería de XML

Localización de servicios

Descripción de servicios

Mensajes orientados a RPC

Mensajes orientados al documento

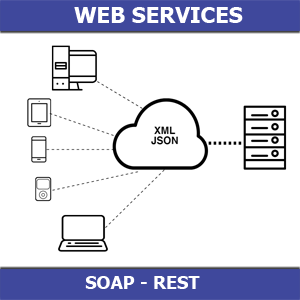


Dentro del mensaje SOAP podemos distinguir los siguientes elementos.

**Servicios web RESTful**

Son un tipo de Servicios web que utilizan estándares Web tales como:

El termino Rest significa REpresentationall State Transfer

****

URls

HTTP

XML

JSON

**Fundamentos RESTful**

Tiene que ser un sistema por capas

Debe que ser sin estado

Debe ser un sistema cliente - servidor

Debe ser un sistema uniformemente accesible

Uso de Hipermedia como maquina de estados de la aplicacion (HATEOAS).

Uso de mensajes auto - descriptivos

Interfaz restringida

Orientado a representaciones

Estar basada en recursos

**Un Servicio Web,** es un **método de comunicación entre dos aparatos electrónicos en una red.** Es una **colección de protocolos abiertos y estándares**usados para **intercambiar datos entre aplicaciones o sistemas.** Las aplicaciones escritas en varios lenguajes de programación que funcionan en plataformas diferentes pueden utilizar web services para intercambiar información a través de una red. La interoperatividad, por ejemplo entre **Java** y **Python** o **Windows** y **Linux** se debe al uso de **estándares abiertos.** (Lázaro, 2018)

**Que es SOAP:**

Los servicios SOAP o mejor conocimos simplemente como Web Services, son servicios que basan su comunicación bajo el protocolo SOAP (Simple Object Access Protocol) el cual este definido por Wikipedia como “protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML”. Por lo tanto, queda claro que la comunicación se realiza mediante XML, lo cual nos debe de quedar muy claro, pues es en este aspecto donde radican las principales diferencias contra REST. Los servicios SOAP funcionan por lo general por el protocolo HTTP que es lo más común cuando invocamos un Web Services, sin embargo, SOAP no está limitado a este protocolo, si no que puede ser enviado por FTP, POP3, TCP, Colas de mensajería (JMS, MQ, etc). Pero como comentaba, HTTP es el protocolo principal. (Blancarte, 2017)

Que es REST

Por otra parte, tenemos REST, el chico nuevo de la cuadra. REST ya tiene unos años, pero en realidad tiene poco que se le empezó a dar la importancia que hoy tiene. REST es una tecnología mucho más flexible que transporta datos por medio del protocolo HTTP, pero este permite utilizar los diversos métodos que proporciona HTTP para comunicarse, como lo son GET, POST, PUT, DELETE, PATCH y a la vez, utiliza los códigos de respuesta nativos de HTTP (404,200,204,409). REST es tan flexible que permite transmitir prácticamente **cualquier tipo de datos**, ya que el tipo de datos está definido por el Header Content-Type, lo que nos permite mandar, XML, JSON, Binarios (imágenes, documentos), Text, etc. que contrasta con SOAP que solo permite la transmisión de datos en formato XML. A pesar de la gran variedad de tipos de datos que podemos mandar con REST, la gran mayoría transmite en JSON por un motivo muy importante, **JSON es interpretado de forma natural por JavaScript**, lo que ha hecho que frameworks como Angular y React se aprovechen al máximo, pues pueden enviar peticiones directas al servidor por medio de AJAX y obtener los datos de una forma nativa. Los formularios de HTML pueden ser apuntados a los servicios REST sin ningún problema (Blancarte, 2017)

**02/09/2020: Arquitecturas de sistemas distribuidos.**

**Arquitectura cliente – servidor:** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados [servidores](https://es.wikipedia.org/wiki/Servidor), y los demandantes, llamados [clientes](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)). Un cliente realiza peticiones a otro programa, el [servidor](https://es.wikipedia.org/wiki/Servidor), quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo [multiusuario](https://es.wikipedia.org/wiki/Multiusuario) distribuido a través de una [red de computadoras](https://es.wikipedia.org/wiki/Red_de_computadoras).

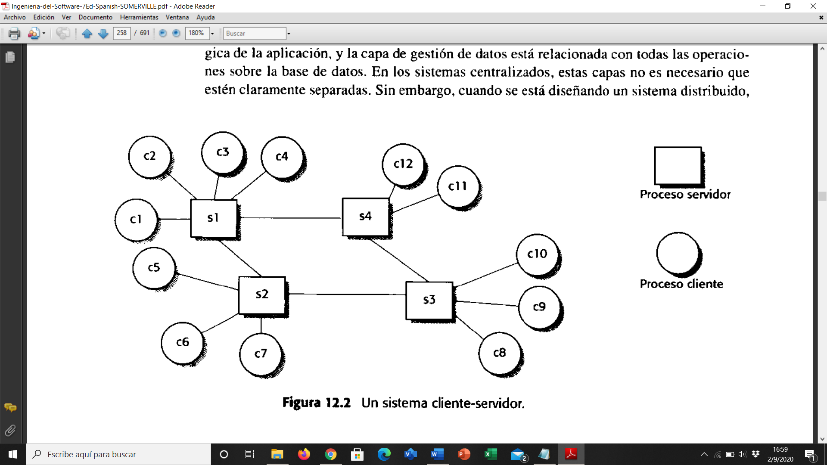
**Arquitectura de objetos distribuidos:** Proporciona un conjunto de servicios que permiten la comunicación entre los objetos y que éstos sean añadidos y eliminados del sistema.

**Arquitectura cliente - servidor**

Clientes y servidores son procesos diferentes.

Modela un conjunto de servicios

Proporcionados por los servidores



**Poseen dos tipos:**

Modelo cliente rico (fat – client)

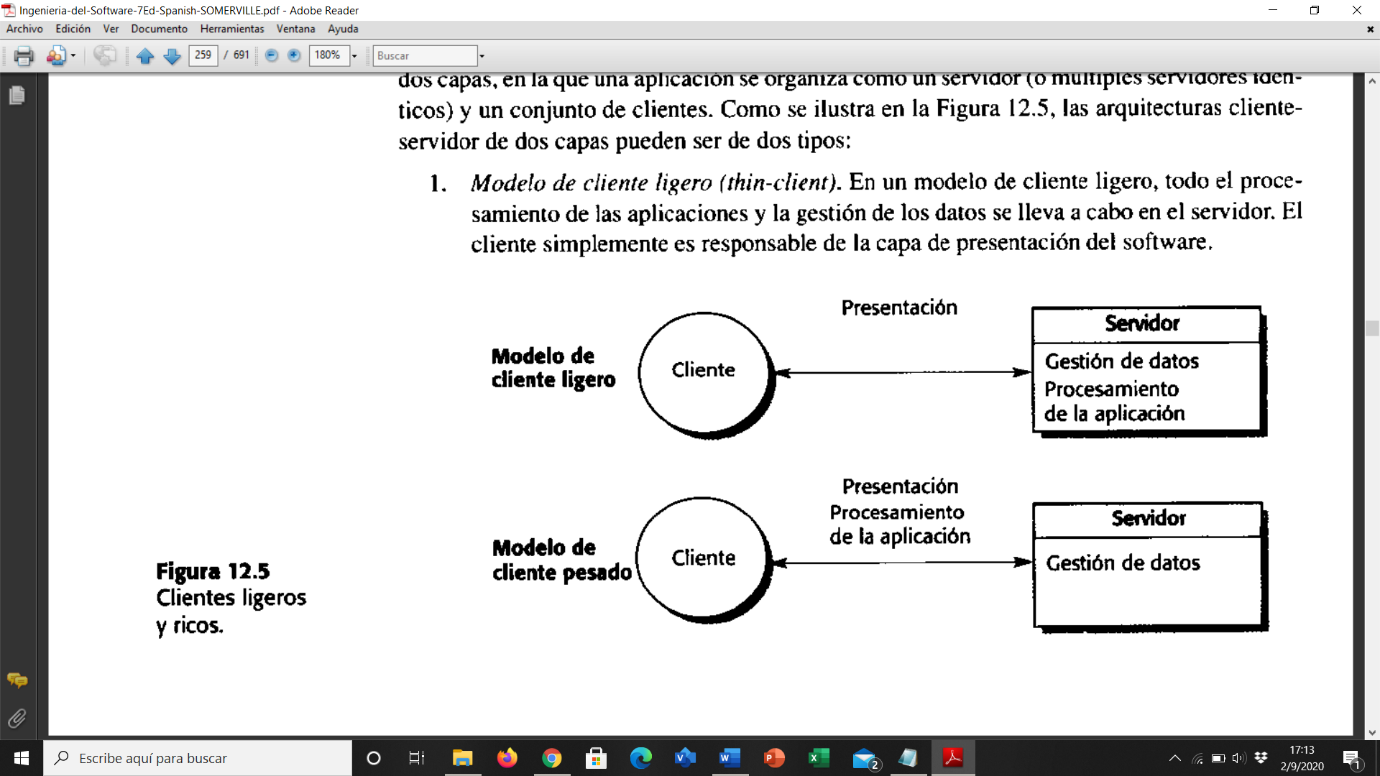
Modelo cliente ligero (thin - client)

El servidor es responsable de la gestión de datos

Todo el procesamiento y gestión de datos

se lleva a cabo

En el servidor



**Arquitectura de objetos distribuidos**

Este modelo funciona para muchos tipos de aplicaciones

Los clientes y servidores son diferentes.

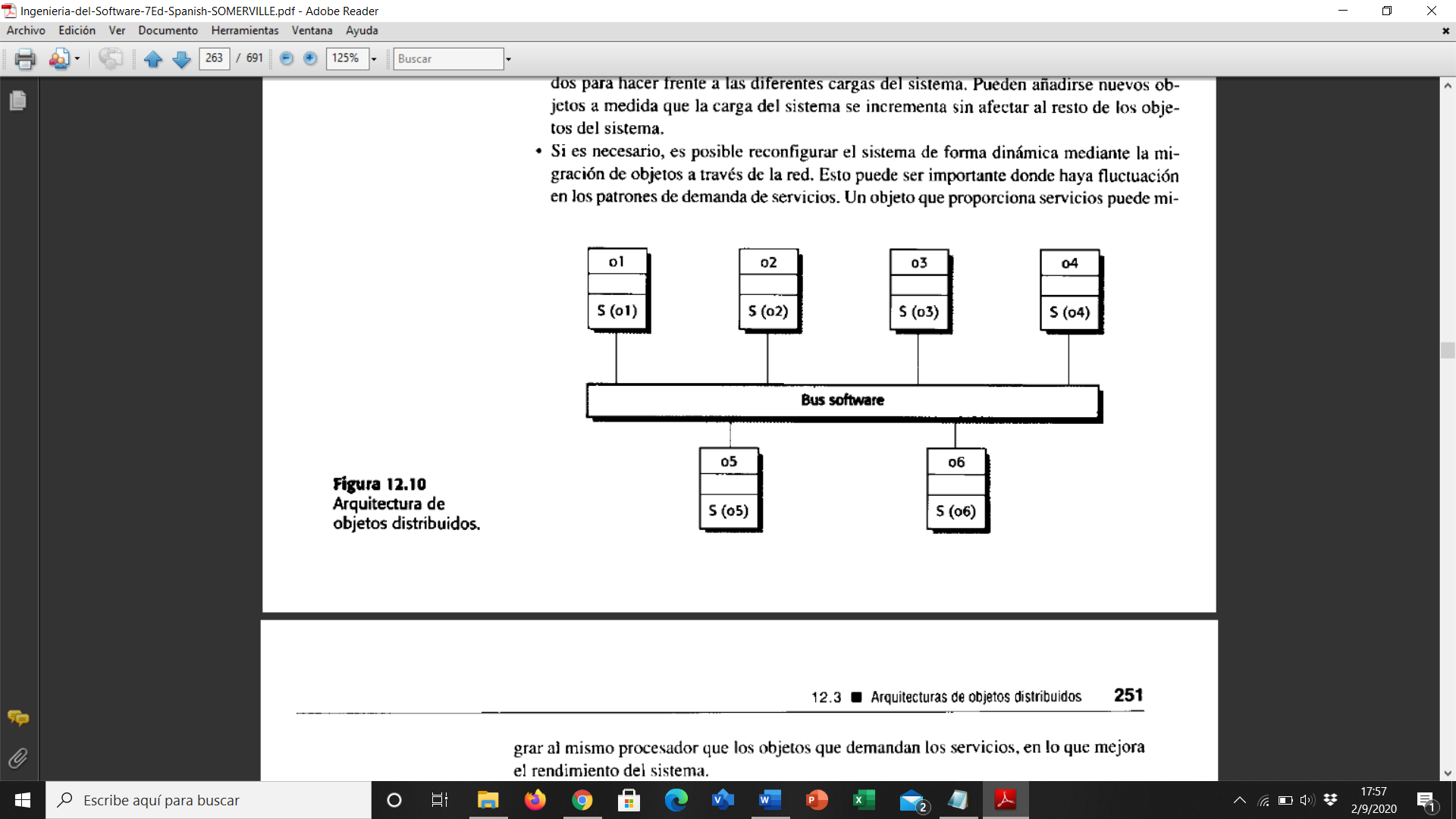
**Ventajas del modelo distribuido**

El sistema es flexible y escalable

Es una arquitectura de sistema abierto

Permite al diseñador retrasar decisiones

Se puede reconfigurar el sistema de forma dinámica



**Componentes de la arquitectura cliente servidor**

Para entender este modelo vamos a nombrar y definir a continuación algunos conceptos básicos que lo conforman.

* **Red:** Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.
* **Cliente:** El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.
* **Servidor:** Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.
* **Protocolo:** Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.
* **Servicios:** Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.
* **Base de datos:** Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes. (Schiaffarino, 2019)

**Arquitectura de sistemas distribuidos**

Un sistema distribuido es un conjunto de ordenadores que trabajan juntos de forma coordinada, a través del intercambio de mensajes, para conseguir un objetivo. En dicho sistema, el estado y los programas se guardan en múltiples ordenadores. A pesar de que los procesos que tienen lugar están separados entre los diferentes participantes, para el usuario parece que está trabajando con un único ordenador. (Rodriguez, 2019)

**07/09/2020: Que es DevOps.**

DevOps es una cultura, un movimiento o practica que enfatiza la colaboración y la comunicación entre los desarrolladores de software y los administradores de sistemas (operaciones) automatizando los procesos de entrega de software y los cambios de infraestructuras de extremo a extremo.

**DevOps**

**Que es DevOps**

**Que no es DevOps**

Un movimiento

Es una cultura

No es una persona

No es un puesto

No son solo herramientas

No es una metodología

Una practica

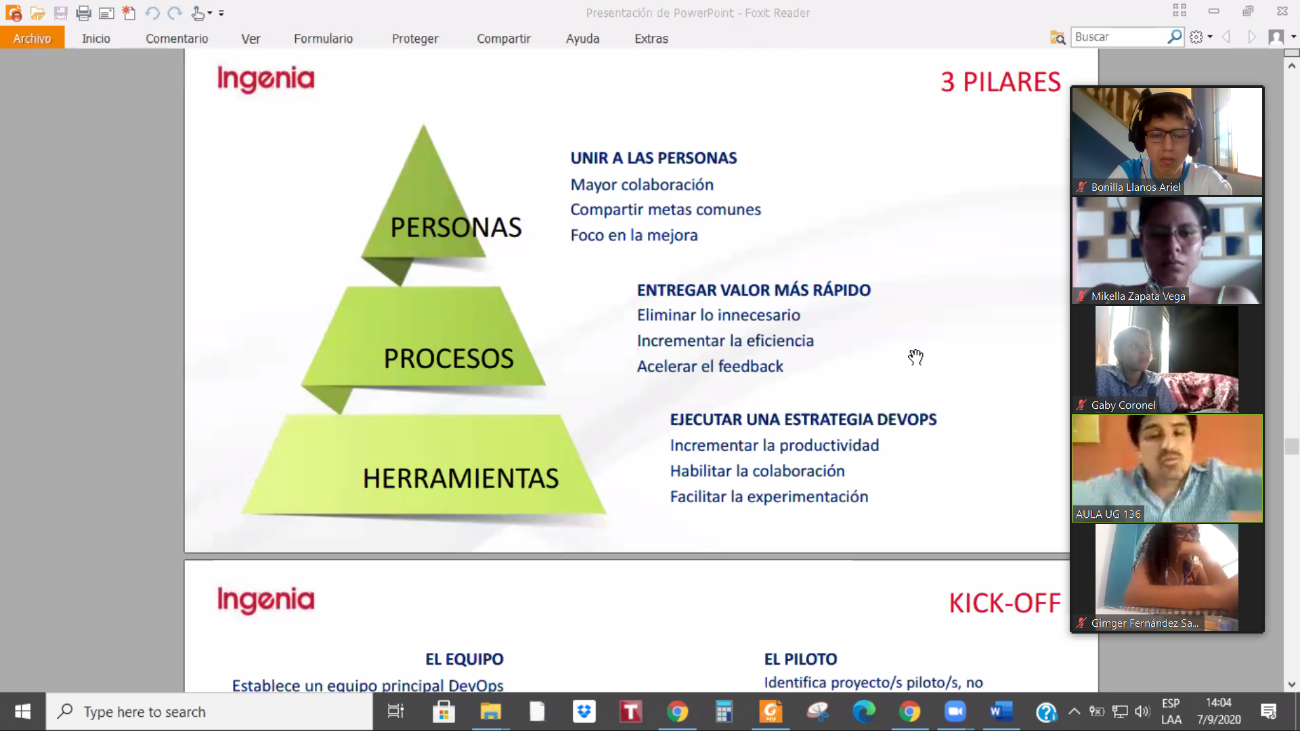


No es solo automatizar

**Que es Devsecops**

Es velocidad de reléase.

Orientación al rendimiento



Cultura

Automatización

Métricas

Apoyarse

Compartiendo

**DevOps**

DevOps es una metodología de trabajo que focaliza en la**comunicación, colaboración**e**integración entre desarrolladores de software y el resto de los profesionales TIC.** El objetivo es ayudar a una organización a producir productos y servicios software más rápidamente, de mejor calidad y a un coste menor. El lógico que todos hablen de DevOps en el sector TIC. (Riesgo, 2019)

**¿Qué hace un DevOps?**

Es un profesional que se encarga de la codificación, la reingeniería de procesos y la comunicación con el resto de las áreas de la empresa. Debe de ser un técnico con experiencia en administración ágil de sistemas y la colaboración con los equipos de trabajo.

Se necesitan profesionales especializados en impulsar el cambio en la cultura TIC, que mejoren la colaboración entre las operaciones y los equipos de desarrollo.  Para ello, se utilizan tecnologías que pueden aprovechar una infraestructura cada vez más programable y dinámica.

# **14/09/2020: Gif Lab.**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

**Sistema de control de versiones.**

**¿Qué es?**

**Ventajas**

“Software para gestionar el historial de versiones de un proyecto”.

Deshacer cambios.

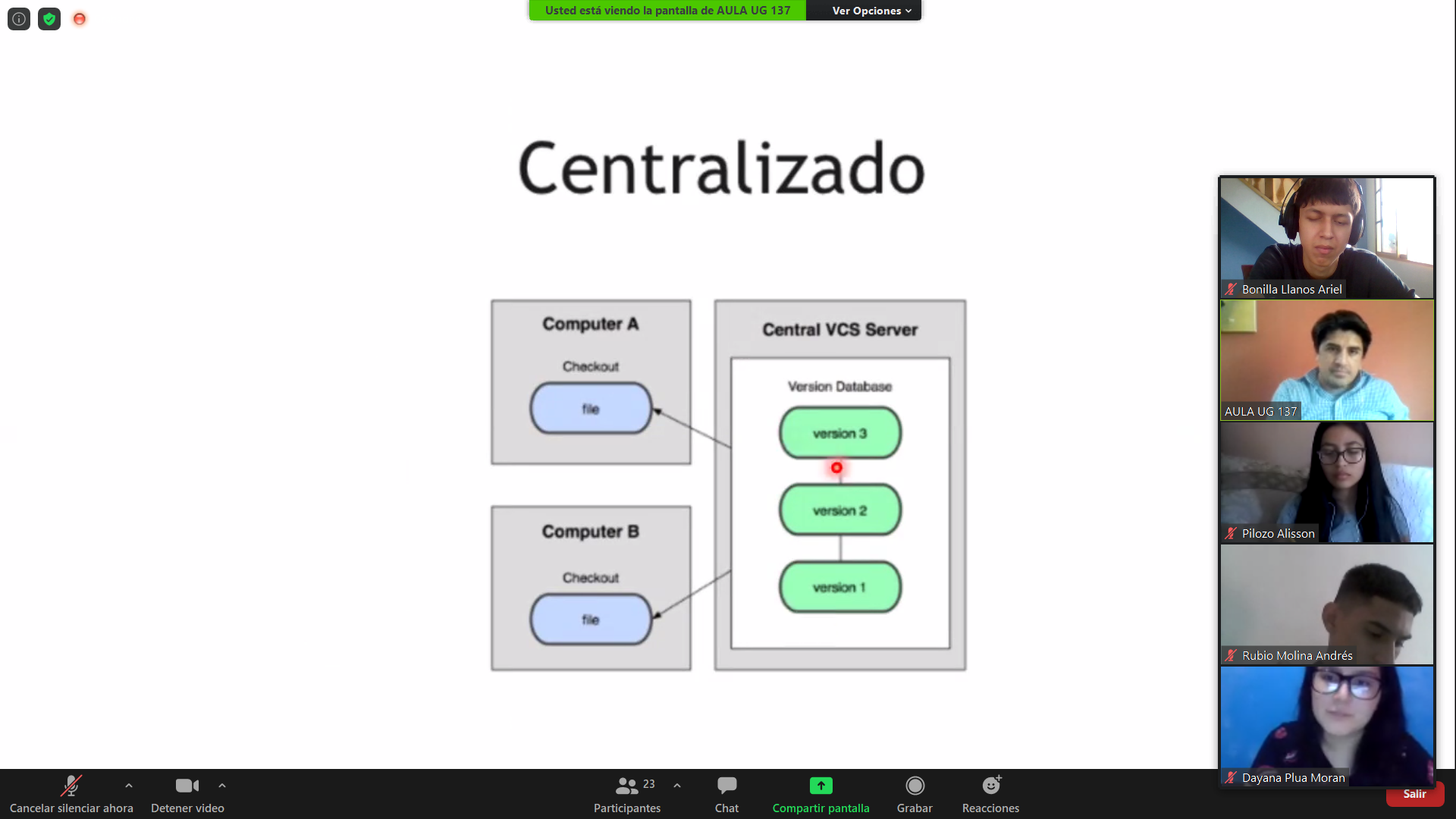
Historial de cambios.

Diferentes versiones del proyecto.

**¿En que tipos de proyectos?**

Trabajos, PFC, tesis, documentación, traducciones, software

**Centralizado**



**Git**

**Ventajas**

**Definición**

* Local
* Pequeño
* Limpio
* GitHub
* Distribuido
* Rápido
* Eficiente
* Seguro
* Flexible

Es un sistema de control de versiones pensado para proyectos.

**Desventajas**

* Curva de aprendizaje.
* Numero de comandos.
* Significado de comandos.

**Control de Versiones con GIT**

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación. (Arduino, 2020)

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o VCS (del inglés Version Control System). Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). (Arduino, 2020)

**Ejemplo** sobre este tipo de herramientas son: [CVS](https://es.wikipedia.org/wiki/CVS), [Subversion](https://es.wikipedia.org/wiki/Subversion), [SourceSafe](https://es.wikipedia.org/wiki/SourceSafe), [ClearCase](https://en.wikipedia.org/wiki/Rational_ClearCase), [Darcs](https://es.wikipedia.org/wiki/Darcs), [Bazaar](https://es.wikipedia.org/wiki/Bazaar_(software)), [Plastic SCM](https://es.wikipedia.org/wiki/Plastic_SCM), [Git](https://es.wikipedia.org/wiki/Git), [SCCS](https://es.wikipedia.org/w/index.php?title=SCCS&action=edit&redlink=1), [Mercurial](https://es.wikipedia.org/wiki/Mercurial), [Perforce](https://es.wikipedia.org/wiki/Perforce), [Fossil SCM](https://es.wikipedia.org/wiki/Fossil_(gesti%C3%B3n_de_configuraci%C3%B3n_de_software)), Team Foundation Server.

**Glosario**

**Programa:** Es un conjunto de instrucciones.

**Protocolo:** Es un conjunto de reglas e instrucciones que gobiernan la comunicación en un sistema distribuido, es decir, el intercambio de mensajes.

**Proceso:** Es un programa en ejecución.

**Interoperabilidad:** Es la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos.

**Extensibilidad:** La cualidad, condición, índole o característica de extensible, que se puede extender o que tiene la probabilidad o posibilidad de se extendido, en ampliar, ensanchar o aumentar de tamaño o superficie, que se esparce o desparrama lo que esta espeso o amontonado.

**Acoplarse:** Adaptarse [una persona] a algo, acomodarse o encontrar acomodo.

**Servicios:** Utilidad o función que desempeña una cosa.

**Hipermedia:** Ve para designar al conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que integren soportes tales como: [texto](https://es.wikipedia.org/wiki/Texto), [imagen](https://es.wikipedia.org/wiki/Imagen), [video](https://es.wikipedia.org/wiki/Video), [audio](https://es.wikipedia.org/wiki/Audio), [mapas](https://es.wikipedia.org/wiki/Mapa) y otros [soportes de información emergentes](https://es.wikipedia.org/w/index.php?title=Soportes_de_informaci%C3%B3n_emergentes&action=edit&redlink=1), de tal modo que el resultado obtenido, además, tenga la posibilidad de interactuar con los usuarios.

**Retrasar:** Situar algo o a alguien en una posición que está más atrás.

**Escalable:**  Es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

**Dinámica:** Parte de la física que estudia el movimiento en relación con las causas que lo producen.

**Automatizar:** Aplicar máquinas o procedimientos automáticos en la realización de un proceso o en una industria.

**Métricas:** Medida, estructura y combinación de los versos de una determinada composición poética, de un escritor, de una época o de un lugar.

**Reléase:** Es una versión que se distribuye a los clientes. Cada reléase incluye nueva funcionalidad o está concebida para diferentes plataformas de hardware. Un reléase del sistema no es sólo el código ejecutable del sistema.

**Archivos:** Un archivo o fichero informático es un conjunto de bytes que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.

**Gestionar:** Hacer las gestiones necesarias para conseguir o resolver una cosa.

**Revisión:** Análisis o examen atento y cuidadoso de una cosa.

# **Bibliografía**

Arduino. (01 de 09 de 2020). *Aprendiendo Arduino*. Obtenido de Control de Versiones con GIT: https://aprendiendoarduino.wordpress.com/2019/09/01/control-de-versiones-con-git/

Blancarte, O. (06 de 03 de 2017). *Oscar Blancarte software architect*. Obtenido de SOAP vs REST ¿cual es mejor?: https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/

Campos, M. (17 de 07 de 2019). *TodaMateria*. Obtenido de Justificación de un proyecto: https://www.todamateria.com/justificacion-de-un-proyecto/

Lázaro, D. (27 de 04 de 2018). Obtenido de Introducción a los Web Services: https://diego.com.es/introduccion-a-los-web-services

Losada, S. (31 de 07 de 2018). *OpenWebinars*. Obtenido de Qué es un sistema distribuido y qué ventajas aporta su funcionamiento: https://openwebinars.net/blog/que-es-un-sistema-distribuido/

Riesgo, J. (28 de 05 de 2019). *RRHHDigital*. Obtenido de ¿Qué es DevOps y que debes saber para convertirte en ello?: http://www.rrhhdigital.com/secciones/tecnologia-e-innovacion/136859/Que-es-DevOps-y-que-debes-saber-para-convertirte-en-ello?target=\_self

Rodriguez, A. (30 de 09 de 2019). *medium*. Obtenido de ¿Qué es un sistema distribuido?: https://medium.com/@adr.rod87/qu%C3%A9-es-un-sistema-distribuido-442870a40bc4

Schiaffarino, A. (12 de 03 de 2019). *infranetworking*. Obtenido de Modelo cliente servidor: https://blog.infranetworking.com/modelo-cliente-servidor/