**DISEÑO Y PROGRAMACION WED**

Actividad Académica

Investigar sobre acciones que se realiza con un versionador de código:  
Explica ¿En qué consiste un...?

1. Commit
2. Push
3. Pull
4. Branch
5. Fetch
6. Merge

#### **Criterios a evaluar**

1. **Aplica reglas ortográficas (1p)**
2. **Coherencia en las ideas (1p) (Caso contrario se anula todo el trabajo)**
3. **Emite juicio crítico(1p)**
4. **Realiza el trabajo en el formato establecido: repositorio\_nombreApellido.docx (1p)**
5. **Realiza un Commit del trabajo en el repositorio correspondiente (1p)**

**Total de puntos: 5**

# **Commit**

**Consolidar**, **confirmar** o **hacer un *Commit*** se refiere, en el contexto de la [ciencia de la computación](https://es.wikipedia.org/wiki/Ciencia_de_la_computaci%C3%B3n) y la [gestión de datos](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_datos), a la idea de confirmar un conjunto de cambios provisionales de forma permanente. Un uso popular es al final de una [transacción de base de datos](https://es.wikipedia.org/wiki/Transacci%C3%B3n_de_base_de_datos).

Una sentencia COMMIT en [SQL](https://es.wikipedia.org/wiki/SQL) finaliza una [transacción de base de datos](https://es.wikipedia.org/wiki/Transacci%C3%B3n_de_base_de_datos) dentro de un  (RDBMS) y hace visibles todos los cambios a otros usuarios. El formato general es emitir una sentencia BEGIN WORK o BEGIN TRANSACTION (o la que sea para el lenguaje SQL en cuestión), una o más sentencias SQL, y entonces la sentencia COMMIT

**PUSH**

Los servicios *Push* están basados, a menudo, en preferencias de información a medida. Es decir, un modelo publicación-suscripción. Un cliente deberá suscribirse a varios canales de información. Cuando el nuevo contenido está disponible en uno de estos canales, el servidor deberá *publicar* (enviar) la información en el dispositivo del usuario.

Las [conferencias sincronizadas](https://es.wikipedia.org/w/index.php?title=Conferencias_sincronizadas&action=edit&redlink=1) y la [mensajería instantánea](https://es.wikipedia.org/wiki/Mensajer%C3%ADa_instant%C3%A1nea) son ejemplos típicos de los servicios tipo *push*. Los mensajes de chat y, en ocasiones archivos, son enviados al usuario tan pronto estos son recibidos por el sistema de mensajería. Los programas descentralizados [P2P](https://es.wikipedia.org/wiki/Peer-to-peer) (como [WASTE](https://es.wikipedia.org/wiki/WASTE)) y los centralizados (como [IRC](https://es.wikipedia.org/wiki/Internet_Relay_Chat) o [Jabber](https://es.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol" \o "Extensible Messaging and Presence Protocol)) permiten hacer «push» de archivos. Es decir, el remitente inicia la transferencia de datos, en vez del destinatario.

El [correo electrónico](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico) es un sistema push: el protocolo [SMTP](https://es.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) en el que está basado, es un protocolo push (ver [Push correo electrónico](https://es.wikipedia.org/w/index.php?title=Push_correo_electr%C3%B3nico&action=edit&redlink=1)). Sin embargo, el último paso: desde el servidor de correo electrónico, hasta el escritorio del usuario usa un protocolo *pull* como [POP3](https://es.wikipedia.org/wiki/Post_Office_Protocol) o [IMAP](https://es.wikipedia.org/wiki/Internet_Message_Access_Protocol). Los clientes modernos de correo electrónico hacen que este paso parezca instantáneo (hacen uso de consultas periódicas al servidor de correo, buscando nuevos mensajes). El protocolo IMAP incluye el comando [IDLE](https://es.wikipedia.org/wiki/IMAP_IDLE), que permite al servidor comunicarle al cliente que ha llegado un nuevo correo. La [BlackBerry](https://es.wikipedia.org/wiki/BlackBerry) original fue el primer ejemplo popular de tecnología push en un contexto inalámbrico.

Otro ejemplo popular de tecnología push en Internet fue [PointCast Network](https://es.wikipedia.org/w/index.php?title=PointCast_(dotcom)&action=edit&redlink=1" \o "PointCast (dotcom) (aún no redactado)), que ganó popularidad en los años 90. Este se encargaba de repartir y guardar datos. Ambos [Netscape](https://es.wikipedia.org/wiki/Netscape_Communications_Corporation) y [Microsoft](https://es.wikipedia.org/wiki/Microsoft) lo integraron en sus programas durante la [guerra de navegadores](https://es.wikipedia.org/wiki/Guerra_de_navegadores), pero fue reemplazado más tarde, en los 2000 por [RSS](https://es.wikipedia.org/wiki/RSS) (una tecnología *pull*).

Otros usos de [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) que hacen uso de la tecnología *push*, incluye a la distribución de datos de mercados (inventarios), sistemas de mensajería/[chat](https://es.wikipedia.org/wiki/Chat) en línea, subastas, juegos y casinos en línea, resultado de deportes, monitorización de consolas y de [redes](https://es.wikipedia.org/wiki/Topolog%C3%ADa_de_red).

**PULL**

**¿QUÉ ES?**

Cuando la demanda del producto determina cuánto producir, se habla de **sistema pull o enfoque pull**. Los tamaños de las órdenes de producción son pequeños, se generan **bajos costes por inventarios, y un riesgo bajo por obsolescencia del producto**.

Este enfoque es conveniente cuando se compite por innovación y flexibilidad, y su implantación requiere de información rápida desde los puntos de venta, así como de un **sistema de producción rápido y flexible**. Las desventajas de este enfoque son la necesidad de tener capacidad para los períodos de demanda pico, menores economías de escala y transporte.

**¿CÓMO FUNCIONA?**

Los distribuidores determinan individualmente las necesidades de reposición de su stock calculando la cantidad requerida y cursando el pedido al almacén regulador. Las **ventajas** de este sistema son las siguientes:

* Mejor conocimiento del mercado.
* Proximidad al cliente por parte de la delegación.

## ****¿Qué es SD Branch?****

SD Branch es la fusión de redes definidas por software con arquitecturas de red para mejorar y simplificar las redes de sucursales. En la actualidad, las redes de sucursales modernas están repletas de hardware y componentes que, gracias a SDN, ya no son necesarios en esta iteración.

En los últimos años, la necesidad de hardware de aplicación propietario ha disminuido gradualmente y esta eliminación y reemplazo con software virtual que se ejecuta en dispositivos estándar disponibles no es más que una de las formas en las que SD Branch busca simplificar la arquitectura de la red de sucursales. Un solo dispositivo SD-WAN es capaz de reemplazar múltiples funciones de hardware dentro de la red de una sucursal específica, sin embargo, la sucursal SD busca la convergencia de SD-WAN con otros elementos tales como conmutadores Wi-Fi, de seguridad y Ethernet.

Ahora bien, esto no significa que el hardware desaparecerá por completo, estas redes seguirán necesitando hardware físico para ejecutarse desde una ubicación específica de la sucursal; sin embargo, sí significa que la reducción de los costes de inversión y mantenimiento de la arquitectura de la red podría, como resultado, impulsar una mayor inversión en SD-WAN, SD-Branch y SDN en general.

Junto a la virtualización de los dispositivos de hardware se ha producido un cambio en la forma en que las empresas están operando, con un porcentaje significativo que ahora se decanta por el software como servicio (SaaS) y las aplicaciones y la gestión basadas en la nube, lo que, a su vez, ha supuesto una carga adicional para las redes de sucursales. En las sucursales de SD, la gestión está automatizada y centralizada, al igual que el marco de control, lo que permite un servicio similar al de la nube.

**FETCH**

Fetch es un nuevo API (ya no tan nuevo) para el acceso a recursos del servidor de manera asíncrona, basado en promesas. Es básicamente la nueva interfaz para realizar funcionalidades Ajax con Javascript, que ya podemos usar para facilitar la organización del código en nuestras aplicaciones.

El acceso a recursos de un servidor de manera asíncrona, comúnmente llamado Ajax, nos permite realizar solicitudes HTTP sin necesidad de recargar toda la página. Tradicionalmente se vienen usando diversos mecanismos para esta tarea, basados en APIs de los navegadores que pueden tener diferencias entre distintos clientes web. Es por ello que muchas veces acabamos usando una librería como [jQuery para acceder a las funcionalidades de Ajax](https://desarrolloweb.com/articulos/uso-ajax-jquery.html).

Fetch ofrece una nueva interfaz estándar de uso de Ajax para el desarrollo frontend, a la vez que permite usar promesas, que nos facilitan la organización del código asíncrono en las aplicaciones. Es un mecanismo disponible actualmente en todos los navegadores, exceptuando los viejos Internet Explorer.

El método fetch() depende directamente del objeto window del navegador. Su uso más simple consiste en pasarle una URL, cuyo contenido se traerá el cliente web de manera asíncrona.

## Merge

El **Merge** o ordenamiento por mezcla, es un ordenamiento con una complejidad computacional logarítmica O(nlog{n})*O*(*nlogn*), que permite ordenar un listado de elementos de manera muy rápida. Este algoritmo, es el primero de los algoritmos de complejidad logarítmica que se mostrarán en el sitio.

Una de las características con las que cuenta el **Merge**, es que es un ordenamiento estable. Los ordenamientos estables, son aquellos que una vez que se ha terminado de ordenar el listado, todos los elementos iguales, quedan en el mismo orden en que originalmente estaban en la lista no ordenada. A continuación se muestra un ejemplo:

**3** 2 **1** 1 **2** 3

**1** 1 2 **2** **3** 3