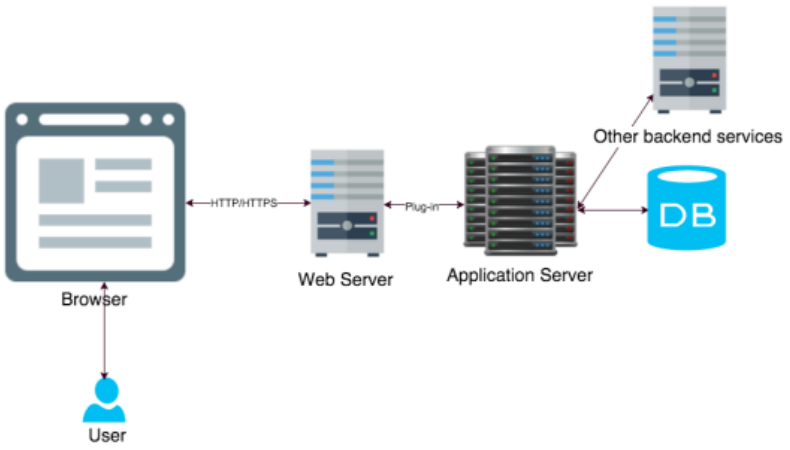
TEMA 3: Servidores de aplicaciones. Despliegue de aplicaciones.

1. INTRODUCCIÓN:

-**Servidor de aplicaciones** → marco mixto de software que permite tanto la creación de aplicaciones web como un entorno de servidor para ejecutarlas. Se sitúa entre el servidor web y el nivel de backend del servidor de BD (es el intermediario para el usuario y el servidor de BD). Es normal que se utilice con un servidor web o lo contenga (así se denomina servidor de aplicaciones web).



2. SERVIDORES DE APLICACIONES:

-Las aplicaciones vienen en todos los tamaños, formas y casos de uso. Los serv. de aplicaciones proporcionan recursos de aplicaciones a usuarios y clientes web.

-Se sitúan física o virtualmente entre los servidores de BD que almacenan los datos de aplic. y los servidores web que se comunican con los clientes. Soportan el desarrollo y la entrega de aplicaciones.

-Cuando se solicita acceso a una aplicación, el serv. de aplicaciones hace el trabajo pesado en el backend para almacenar y procesar la solicitud de la aplicación.

¿POR QUÉ NECESITAMOS SERV. DE APLICACIONES?

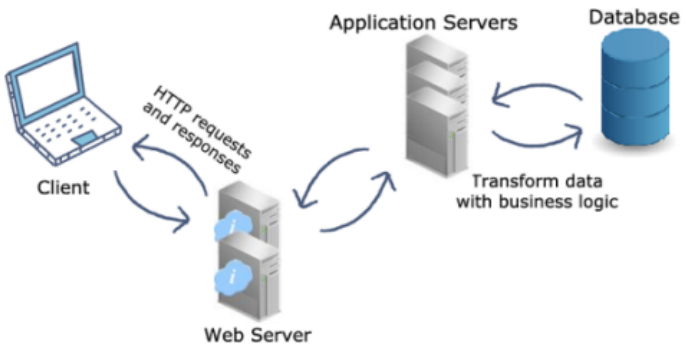
-Los serv. web se encargan de servir a los clientes una respuesta HTTP. A diferencia de los serv. de aplicaciones, el diseño de serv. web es lo suficientemente ligero como para procesar las solicitudes de datos estáticos de varias aplicaciones(o sitios web), manteniendo la seguridad. Las peticiones dinámicas requieren asistencia adicional y aquí, es donde intervienen los **serv. de aplicaciones**, ya que **manejan el contenido web dinámico.**

-Proporcionan también una capa adicional de seguridad, una vez desplegado entre una BD y un servidor web. Las organizaciones pueden proteger más aún con un servidor proxy inverso colocado delante de sus BD.

¿CÓMO FUNCIONAN LOS SERV. DE APLICACIONES?

Los servidores de aplicaciones tienen características de seguridad, transacciones, clustering, diagnósticos y BD. **En lo que se diferencian los servidores web es en su capacidad para procesar peticiones de servlets (programas Java) desde un servidor web.**

Flujo general de los serv. de aplicaciones web:



1. Cliente abre navegador y solicita acceso a sitio web.
2. Servidor web recibe petición HTTP y responde con la página deseada.
3. Servidor web gestiona las peticiones de datos estáticos, pero el cliente quiere utilizar una herramienta interactiva.

4. Al ser una petición de datos dinámicos, el servidor web transfiere la petición a un servidor de aplicaciones.
5. El serv. de aplicaciones recibe la petición HTTP y la convierte en petición servlet.
6. El servlet llega al servidor de BD y el servidor de aplicaciones recibe la respuesta del servlet.
7. El servidor de aplicaciones traduce la respuesta del servlet al formato HTTP para el acceso al cliente.

-Al recibir una solicitud de servlet de un servidor web, el serv. de aplicaciones procesa la solicitud y responde al servidor web mediante la respuesta servlet. Dado que los servidores de aplic. trabajan principalmente con peticiones de lógica de negocio, traduce la respuesta de servlet y pasa una respuesta HTTP accesible al usuario. El servidor web está diseñado para manejar peticiones HTTP, el serv. de aplicaciones maneja otro tipo de peticiones.

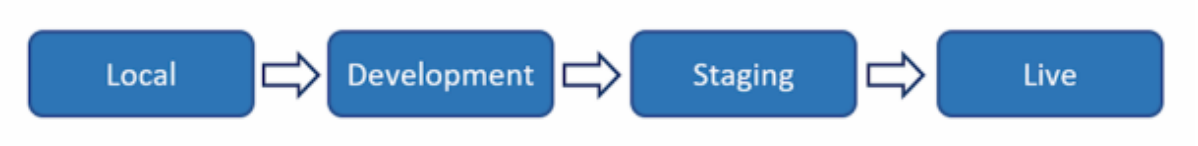
3. DESPLIEGUE DE APLICACIONES:

-El despliegue significa pasar los cambios o actualizaciones de un entorno de funcionamiento a otro. Al configurar un sitio web, siempre se tendrá el sitio web en vivo, que se llama **entorno en vivo o de producción.**

-Si se quiere hacer cambios sin afectar al sitio web en producción, se debe añadir entornos adicionales (**entornos de desarrollo o de despliegue**), que suelen ser un entorno local, un entorno de desarrollo y un entorno de preparación o preproducción. El número de entorno que se necesite depende de cada caso y de la complejidad del proyecto.

-El modelo de despliegue más común es el clásico "de izquierda a derecha" cuando se trabaja con múltiples entornos de despliegue. En este modelo, los cambios se realizan en entornos locales, de desarrollo o de preparación (dependiendo de la configuración) y se van pasando de izquierda a derecha a través de los diferentes entornos, terminando en el de producción. Finalizado el proceso, los cambios son visibles en el **entorno activo**.

-La **ventaja principal** de utilizar múltiples entornos → se pueden hacer cambios sin que afecten a un sitio web en vivo.



3.1 PASOS DEL PROCESO DE DESPLIEGUE:

1. **PLANIFICACIÓN →** al tener un plan nos aseguramos de que todo se haga de la misma manera cada vez que se realicen cambios. Muy útil cuando varios usuarios trabajan en el mismo proyecto. Un plan de despliegue debe incluir **reglas** sobre cuándo desplegar desde los entornos locales a los sitios web de desarrollo o de puesta en escena, así como **horarios** para cuando los nuevos cambios puedan ir a un entorno en vivo. Al tener un plan, se reduce el riesgo de conflictos entre los diferentes cambios y se asegura que el proceso de despliegue sea fácil y fluido. Planificar también los cambios.
2. **DESARROLLO →** establecido el plan, momento del desarrollo. Para que pueda realizarse simultáneamente y sin romper nada, trabajar únicamente en entornos locales o de desarrollo. Realizado el proceso de desarrollo, momento de probar y desplegar los cambios a través de los distintos entornos.
3. **PRUEBAS →** probar es crucial para que no haya error en el entorno de producción final, pero las pruebas no pueden completarse sin desplegar los cambios en nuevos entornos. Comprobado que funcionan los cambios en el entorno local o de desarrollo, es el momento de desplegar los cambios en el siguiente entorno (hasta el entorno de preproducción, donde se realizan las pruebas finales de control de calidad). Si todo es correcto, se despliega en vivo, si no, habrá que volver al entorno de desarrollo a solucionar el error.
4. **DESPLIEGUE →** realizadas las pruebas y corregidos los errores, momento de desplegar los cambios en el entorno real.
5. **SUPERVISIÓN →** es importante revisar que todo funcione según lo previsto. Independientemente de la planificación realizada, existe la posibilidad de que los usuarios encuentren problemas o realicen acciones que no se habían previsto.

- Consejo para la supervisión: planificar los lanzamientos para los momentos en los que haya menos usuarios. El número de usuarios

afectados por cualquier error sería mínimo y se tendrá gente preparada para arreglarlo.

3.2 BUENAS PRÁCTICAS AL REALIZAR EL DESPLIEGUE:

UTILIZAR GIT

Tener un sistema de control de versiones es inestimable para cualquier flujo de trabajo de despliegue, sin él es probable que se produzcan errores si se trabaja en equipo. Inclusive, si eres el único desarrollador que trabaja en un proyecto, es recomendable utilizar Git, en caso de volver a versiones anteriores o por si alguien se une a tu equipo.

Sin Git, es más probable que se cometan más errores al desplegar código inacabado o por no tener a todos los miembros del equipo trabajando en la misma versión.

TRABAJAR EN RAMAS

Trabajar así permite trabajar en varias cosas al mismo tiempo sin que afecten entre sí. Ejemplo de ello, cuando se encuentra un error y debe ser corregido. Si un desarrollador está utilizando una rama para trabajar en una nueva característica, puede hacer rápidamente una nueva rama del entorno de desarrollo para trabajar en el error (habría dos ramas).

UTILIZAR ENTORNO LOCAL COMO ENTORNO DE DESARROLLO

Al instalar el sitio web o software de forma local, se trabaja de forma más eficiente y acelerar las pruebas y verificación del código.

No hay que confirmar, empujar y desplegar constantemente un cambio antes de poder verificar si funciona, y cuando algo no funciona, no se tendrá que revertir, empujarlo y volver a desplegarlo. En lugar de eso, simplemente se ejecuta todo localmente y si funciona, se empuja al entorno de preparación para una prueba más rigurosa.

REVISAR LAS DIFERENCIAS ANTES DE DESPLEGARLO EN EL ENTORNO REAL

Asegurado de que todo funciona en el entorno de pruebas, es el momento de desplegar el código en el entorno real. Antes de ello, es importante hacer una revisión final de las diferencias entre el entorno actual en producción y el entorno de desarrollo del que se parte, por lo que es muy recomendable una revisión final del código antes de pulsar el botón de despliegue.

¿A QUÉ HORA DEL DÍA SE DEBEN DESPLEGAR LOS CAMBIOS?

En caso de que se rompa al desplegar en el entorno de producción,es importante encontrar el mejor momento para hacerlo. Este momento puede variar mucho entre un proyecto y otro. Hay dos preguntas que suelen hacerse para determinar cuándo desplegar los cambios:

> **¿Cuándo tiene la aplicación menos usuarios?**
>
> **¿Cuándo hay alguien que pueda solucionar posibles fallos del nuevo despliegue?**

3.3 VENTAJAS DEL DESPLIEGUE EN ENTORNOS MÚLTIPLES:

REDUCCIÓN DEL RIESGO DE ROMPER UN SITIO WEB EN PRODUCCIÓN

El uso de múltiples entornos y confianza en el despliegue reduce el riesgo de que los cambios tengan un impacto negativo en un sitio web en vivo. Mientras que los cambios menores se pueden hacer fácilmente en un sitio web en vivo directamente, los cambios más grandes se pueden hacer en entornos desamparados sin el riesgo de romper nada en el entorno vivo.

AHORRO DE TIEMPO

Sin la preocupación de romper algo, se pueden realizar los cambios en el orden que se prefiera. Ésto optimiza el flujo de trabajo para realizar los cambios sin tener en cuenta el aspecto o funcionamiento del sitio web mientras se lleva a cabo.

Si se trabaja en entorno local también está la ventaja de que los cambios se procesan más rápido y sin dependencias de conectividad.

A la hora de desplegar, también se ahorrará tiempo, ya que todos los cambios podrán realizarse al mismo tiempo en lugar de tener que hacerlo en varios pasos más pequeños.