Despliegue:

- Etapa del proceso del desarrollo en la que la app/web se publica para que cualquier usuario la pueda utilizar.
- Dependiendo del tipo de producto con el que se trabaje se utilizarán unas herramientas u otras.

Ejemplos Plataformas despliegue:

- **Vercel**: gratuita y fácil de utilizar enfocada al *open source*.
- **Heroku**: ofrece soporte para varios lenguajes de programación.
- **Surge**: Gratuita. Alojar y desplegar aplicaciones estáticas.
- **Netlify**: Opción extraordinaria cuando se trata de acelerar el envío de la app del lado del cliente.

ETAPAS

- **Plan**: Sincronización de desarollo y operaciones, trazando el cronograma de implementación de la app y evaluando la infraestructura actual.
- **Construcción y lanzamiento automático.** Minimizar errores humanos.
- **Desarrollar integración continua:** Reducción del grado de cambio.
- **Probar y crear guiones:** Identificar cambios y diferencias ambientales ejecutando scripts de prueba en una copia de seguridad.
- **Probar**: Verificar que todo funciona bien.
- **Desarrollar seguimiendo despliegue:** Habilitar servicios de seguimiento para que los equipos rastreen cuando suceden los despliegues e identificar errores.
- **Alertar a los usuarios:** Ayuda a coordinar el proceso, establecer expectativas y dar marcha atrás en caso de error.
- **Supervisar y renovar:** Una vez despliegue hecho hay que seguir supervisando que no se produzcan errores e ir corrigiendo todo lo que sea necesario.

Arquitectura Web Elementos

Servidor Web, Internet,Cliente Web

Tipos Arquitectura

Cliente-Servidor
Capas físicas - modelo 3 capas

- ***Presentación (Navegadores Web)***
  Interpretan peticiones del usuario y presentan los resultados al usuario
- ***Proceso (Servidor Web)***
  Controlan presentación
  Operaciones de la app web
  Interactúan con los servidores datos
- ***Datos (Servidor Datos)***
  Servidores BBDD
  Servidores ficheros
  Servidores correo

Protocolos Red

FTP: Intercambio de archivos nosts/local/remotos.
HTTP: Protocolo de Transferencia Hipertexto.
IMAP: Protocolo de correo electrónico.
DNS: Sistema de nombres de dominio (IP)
DHCP: Automatizar la asignación de IPs en la red.

Antes de INSTALACIÓN

Actualizar paquetes SO
sudo apt update
sudo apt upgrade

Conf. sitio web seguro con HTTPS: PASOS

1.Instalar módulo de SSL sudo apt-get install openssl
2.Generar certificado SSL
1. Crear clave: openssl genrsa -out **mi-sitio.key** 2048
2. Generar solicitud certificado:
openssl req -new -key **mi-sitio.key** -out **mi-sitio.csr**
3. Generar certificado autofirmado con clave privada
openssl x509 -req -days 365 -in **mi-sitio.csr** -out **mi-sitio.ctr**
4. Configurar host virtual con HTTPS
1. Configuración del host en: /var/www /mi-sitio.conf
2. Crear en **/etc/apache2/sites-available** archivo mi.sitio.ssl.conf
mi.sitio.ssl.conf habilitar host **sudo a2ensite mi-sitio.**
4. Copiar contenido mi-sitio.conf y añadir lo siguiente:
SSLCertificateFile /etc/ssl/certs/mi-sitio.ctr
SSLCertificateKeyFile /etc/ssl/private/mi-sitio.key
5.Reiniciar apache.

PREPARACIÓN ENTORNO

1.INSTALACION APACHE
sudo apt install apache2
Verificar http://localhost
APACHE http://127.0.0.1
/etc/init.d/apache2 start stop restart

2. INSTALACIÓN PHPMYSQL
sudo apt-get install mysql.-server php-mysql
sudo mysql_secure_installation
sudo mysql -u root -p

3. INSTALACIÓN PHP
sudo apt install php libapache2-mod-php
Verificar PHP
Creamos fichero prueba:/var/www/html /info.php
Accedemos al fichero: http://localhost /info.php

4. INSTALACION PHPMYADMIN
sudo apt-get install phpmyadmin
Verificar:http://localhost/phpmyadmin

5. INSTALACION SERVIDOR FTP
sudo apt install vsftpd
Conf.firewall: sudo ufw allow ftp
sudo ufw allow ftp-data
Conf.ftp: listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
Asignar permisos fichero html
sudo chmod -R 777 /var/www/html

INSTALACIÓN XAMPP BBDD = botigajocs

Verificar v php: php -v **Descargar v compatible con nuestra v php**
Inst. permisos ejecución: sudo su chmod a+x xampp-linux-x64-8.0.30-0-installer.run
Comienza ejecución: sudo ./xampp-linux-x64-8.0.30-0-installer.run
Comprobar los servicios en ejecución: iremos a nuestro phpMyAdmin
Iniciar servidor: sudo /opt/lampp/lampp start
Crear user y pass: CREATE USER 'botigajocs' @'localhost' IDENTIFIED BY 'botigajocs';
Importamos script sql en phpMyAdmin
Eliminar contenido de htdocs (/opt/lampp): rm -r htdocs/*
Copiar contenido directorio bbdd al directorio XAMPP: cp -r /home/esther/Downloads/botigajocs/*.
Abrir navegador web y acceder a localhost: Debe aparecer web botigajocs.

Conf. Hosts virtual: /etc/apache2/sites-enabled /000-default.conf

1.Crear archivo conf. dominio mi-sitio.conf

PASOS
a) copiando 000-default.conf **cp 000-default.conf mi-sitio.conf**
b) crear desde 0.
ServerAdmin webmaster@localhost: email administrador servidor.
ServerName /localhost
DocumentRoot /var/www/html:
COMPUESTO
<Directory var/www/html>
AllowOverride All
</Directory>
archivo que debe aparecer aquí cuyo ServerAlias debe ser igual
ServerAlias /html
ErrorDocument 404 "Error." Indicamos texto que debe aparecer.
/html/error.html Ruta archivo que debe mostrar.
ErrorLog ${APACHE_LOG_DIR}/error.log: ruta archivo sms error.
CustomLog ${APACHE_LOG_DIR}/access.log combined: registro de accesos.

2.Crear directorio /var/www /mi-sitio.conf
ASIGNAR PERMISOS sudo chmod -R 777 /var/www /mi-sitio.conf
sudo chown -R 777 /var/www /mi-sitio.conf

3.Conf. Hosts /etc/hosts
AÑADIR
- ip servidor web (ip a)
- ip mi-sitio.conf

4.Habilitar host: sudo a2ensite mi-sitio **5.Reiniciar apache.**

CONF. APACHE2

Archivo configuración: /etc/apache2/apache2.conf
KeepAlive On: Permite que un cliente pueda reutilizar la misma conexión TCP para realizar múltiples solicitudes.
MaxKeepAliveRequests 100: Define núm max solicitudes permitidas.
KeepAliveTimeout 5: Espera del servidor para una nueva conexión.
AccessFileName: nombre que usa Apache para configurar control de accesos específico (.htaccess por defecto).
Include: Permite mantener conf. separadas por funcionalidad.
Archivo puertos:/etc/apache2/ports.conf
Listen 80
<IfModule ssl_module>
Listen 443
</IfModule>
permite cifrado SSL/TLS necesario para conexiones HTTPS.
<IfModule mod_gnutls.c>
Listen 443
</IfModule>
alternativa al ssl_module que tb permite conexiones seguras.
Archivo web: /var/www /index.html

Restringir acceso directorio APACHE2

1.Creamos archivo contraseñas
sudo htpasswd -c /etc/apache2/.htpasswd **nombreusuario**
2.Configurar .htaccess (añadir)
AuthType Basic tipo encriptación
AuthName Area Restringida sms cuando no deja entrar
AuthUserFile /etc/apache2/.htpasswd sitio donde consigue pass
Require valid-user
3.Configurar archivo configuración.
SI TENEMOS DOMINIO mi-sitio-conf
1. entramos y buscamos <Directory>
2. Añadimos *AllowOveride All*
ELSE entramos en apache.conf . Repetimos 1 y 2.
4.Reiniciar apache.