

Ambientes de producción y tipos de paradigmas que los soportan

Emmanuel Pineda

Miguel Ángel Rico García - 20191020107

miaricog@correo.udistrital.edu.co

Laura Catalina Preciado Ballen -20182020122

lcpreciado@correo.udistrital.edu.co

Resumen— En este documento se dan a conocer los rasgos fundamentales de los ambientes de producción de software y los paradigmas que lo soportan, se hablará que son, los tipos y para que se usan. Se expondrá qué es una tarea y se llegara a demostrar cómo gracias a la aplicación del concepto podremos trabajar en proyectos grandes de software; además de las ventajas que representa trabajar con estos.

Palabras Clave: entorno de desarrollo, niveles del entorno de desarrollo, paradigma, paradigma imperativo, paradigma declarativo.

I. INTRODUCCIÓN

Debido al gran tamaño de los proyectos que puede llegar a manejar una empresa y las continuas actualizaciones se requiere un proceso ordenado y con niveles para el rápido desarrollo del software y adecuado mantenimiento.

II. DESARROLLO DE CONTENIDO

¿Qué son los ambientes de producción?

Un entorno de desarrollo en el mundo del software y la tecnología es un conjunto de procedimientos y herramientas utilizadas por los desarrolladores para codificar, generar, depurar, actualizar, integrar, testear, validar y ejecutar programas.[1]

Actualmente nos referimos a un entorno de desarrollo al proceso integral de gestión del desarrollo del software[1]

Niveles del entorno de desarrollo

1. Entornos de integración continua

Primero se parte de que cada desarrollador tiene un entorno funcionando en distintas máquinas o servidores por los que pasa antes de ser instalado en el servidor real.[1]

En este paso se busca:

- integrar el trabajo en un repositorio central, dando como resultado una

versión del código actualizada y consolidada.

- Automatizar las pruebas de integración y su validación antes de ser movido al siguiente entorno.
- Enviar el código al siguiente entorno si las pruebas han sido superadas satisfactoriamente.[2]

2. Entorno de prueba

Un entorno de pruebas es un espacio seguro en el que, si cometemos un fallo, este no tendrá ninguna repercusión, aislando los cambios en el código.

Los diferentes tipos de entornos de prueba están determinados por las configuraciones del entorno para una determinada prueba. Por ejemplo: un entorno de prueba para Firefox es una configuración del entorno navegador. [3]

3. Entorno de pre-producción

El código será movido al entorno de pre-producción. Aquí se realizarán las pruebas de validación al conjunto del software, teniendo como objetivo localizar cualquier error antes de llegar al entorno de producción y evitar así los problemas derivados de ellos [2]

Este entorno será completamente funcional a nivel de usuario, y si hemos recomendado que el entorno de desarrollo fuese lo más similar posible al entorno de producción, aquí se convierte en algo crítico. Tanto el software, dónde no solo hablamos de las aplicaciones sino también de sus versiones y configuraciones, como el hardware y los conjuntos de datos. Cuanto mayor sea la similitud con el entorno de producción menor será el número de incidencias que nos encontremos cuando el software esté en productivo.[2]

4. Entorno de producción

Dentro del entorno de producción vamos a encontrar que la principal característica de este será que el producto final de software finalmente

será puesto en un escenario de despliegue, quedando ya habilitado para que los usuarios objetivo puedan consumir el aplicativo.

Lo anterior incide directamente dentro de la necesidad de poseer una infraestructura estable y una **mayor capacidad de manejo de tráfico o de conexiones recurrentes**[1], con el fin de facilitar y mejorar la experiencia del usuario y el rendimiento del aplicativo.

Dentro de los entornos de producción podemos encontrar que el paradigma imperativo hacer parte importante de los procesos requeridos para poder desplegar un proyecto, ya se trate de una lógica que corresponde a componentes interrelacionados para crear una interfaz de usuario (Frontend), o por el contrario una lógica para la conexión y el consumo de una API y/o base de datos (Backend).

```
Deploy log
1 6:14:52 PM: Build ready to start
2 6:14:53 PM: build-image version: 653805ca4a64301556e56dc4b321ef8fc20cbb7c
3 6:14:53 PM: build-image tag: v3.8.2
4 6:14:53 PM: buildbot version: 27121fae4d1b3a5c621e9bbd5ca8c7f99e19f330
5 6:14:54 PM: Fetching cached dependencies
6 6:14:54 PM: Starting to download cache of 92.4MB
7 6:14:54 PM: Finished downloading cache in 800.445353ms
8 6:14:54 PM: Starting to extract cache
9 6:14:58 PM: Finished extracting cache in 3.727890967s
10 6:14:58 PM: Finished fetching cache in 4.551281817s
11 6:14:58 PM: Starting to prepare the repo for build
12 6:14:59 PM: Preparing Git Reference refs/heads/master
13 6:14:59 PM: Parsing package.json dependencies
14 6:14:59 PM: Different publish path detected, going to use the one specified
    in the Netlify UI
15 6:15:00 PM: Starting build script
16 6:15:00 PM: Installing dependencies
17 6:15:00 PM: Python version set to 2.7
18 6:15:00 PM: Started restoring cached node version
19 6:15:02 PM: Finished restoring cached node version
20 6:15:03 PM: v12.18.0 is already installed.
21 6:15:04 PM: Now using node v12.18.0 (npm v6.14.4)
22 6:15:04 PM: Started restoring cached build plugins
23 6:15:04 PM: Finished restoring cached build plugins
24 6:15:04 PM: Attempting ruby version 2.7.2, read from environment
25 6:15:05 PM: Using ruby version 2.7.2
26 6:15:05 PM: Using PHP version 5.6
27 6:15:05 PM: Started restoring cached node modules
28 6:15:05 PM: Finished restoring cached node modules
29 6:15:05 PM: Started restoring cached go cache
30 6:15:05 PM: Finished restoring cached go cache
31 6:15:05 PM: go version go1.14.4 linux/amd64
32 6:15:05 PM: go version go1.14.4 linux/amd64
33 6:15:05 PM: Installing missing commands
34 6:15:05 PM: Verify run directory
35 6:15:06 PM:
36 6:15:06 PM:
37 6:15:06 PM: Netlify Build
38 6:15:06 PM:
```

Figura 1. Secuencia de instrucciones para proceso de despliegue en la plataforma Netlify
Autoría propia

Como podemos observar en la figura 1, dentro del proceso de despliegue de un proyecto se requiere de una secuencia de acciones y de verificaciones en orden para poder llevar a cabo la disposición en los servidores de unos archivos estáticos que servirán para el despliegue de una SPA (Single Page Application).

III. CONCLUSIONES

IV. REFERENCIAS

- [1]«ekon.es» 2019. [En línea]. Available: <https://www.ekon.es/entornos-desarrollo-software/>
- [2]«ahierro.es» 2018. [En línea]. Available: <https://blog.ahierro.es/entornos-en-el-desarrollo-de-software/>
- [3]«IBM» 2015. [En línea]. Available: <https://www.ibm.com/docs/es/elm/6.0.1?topic=environments-test-environment-types>