



DESPLIEGUE DE APLICACIONES WEB

Apuntes 2 Evaluación

Descripción breve

Apuntes de la asignatura de Despliegue de Aplicaciones Web del 2 curso del grado superior
DAW

Raquel Rodríguez
Raquelrodrim2000@gmail.com

Contenido

| | |
|---|----------|
| CONTROL DE VERSIONES AVANZADO – GIT | 2 |
| RAMAS | 2 |
| Pasos a seguir para hacer una modificación: | 2 |
| COMANDOS PARA TRABAJAR CON RAMAS | 2 |
| MERGE | 3 |
| Usos de merge | 3 |
| DESPLIEGUE BÁSICO EN SERVIDOR | 5 |
| Comandos | 5 |
| Gestión de imágenes | 5 |
| Gestión de contenedores | 5 |
| Gestión de redes | 5 |

CONTROL DE VERSIONES AVANZADO – GIT

RAMAS

- develop: Lo que se va desarrollando y ya funciona. Cuando se quiera se pasa a release.
- release: Lo que se está probando en un entorno similar al de producción. Cuando está probado, se copia a master.
- master: Lo que ya se puede instalar en la producción.
- rama de funcionalidad: Cada vez que alguien quiere hacer algo, crea una rama desde develop y cuando acaba la fusiona en develop.

Pasos a seguir para hacer una modificación:

- Crea la rama feature desde develop.
- Realizar la tarea con tantos commits como sea necesario.
- Mergear la rama develop en feature ya que así se resuelven los problemas de mergeo en la rama feature
- Ir a la rama develop y mergear la rama feature con --squash
- Borrar la rama feature
- Subir la rama develop
- Mergear la rama develop en release y subir release
- Mergear la rama release en master y subir master

COMANDOS PARA TRABAJAR CON RAMAS

- Crear una rama y subirla a git:
git branch nuevaRama
git switch nuevaRama
git push -u origin nuevaRama
- Bajar de git una nueva rama
git fetch --prune
git switch nuevaRama

Se pone **--prune**, para que borre las referencias locales de ramas remotas que ya no están. Por defecto al cambiar a la nueva rama ya se hace un **--track** a **origin/nuevaRama**

- Borrar una rama en local que ya ha sido mergeada o subida
`git branch -d miRama`
`git branch --delete miRama`
- Borrar una rama en local que NO ha sido mergeada ni subida (hay 3 formas distintas). Pero es usar el modificador -f o --force
`git branch -d -f miRama`
`git branch --delete --force miRama`
`git branch -D miRama`
- Borrar una rama en remoto
`git push origin --delete miRama`

MERGE

El merge es unir los commit de una rama en otra. Hay muchas formas de hacer un merge , veamos algunas de ellas.

| Comando | Características | Cuando usarlo |
|---------------------------------------|---|---|
| <code>git rebase rama</code> | Pone los commits en la nueva rama siempre después de los que ya hay, es decir que los reordena | Poner los commits de origin/develop en nuestra develop porque hay commits distintos en cada rama pero del mismo commit origen |
| <code>git merge --squash rama</code> | Junta todos los "microcommits" de la rama en uno solo en la rama destino. Es necesario acto seguido el commit | De una rama feature a develop |
| <code>git merge --ff-only rama</code> | Nunca crea nuevos commits en la rama destino pero falla si es necesario crear un nuevo commit, por lo que <code>--ff-only</code> ayuda a detectar si hay algún problema | De develop a Release De release a master o para actualizar cualquier rama desde GitHub cuando no hay commits nuevos en local |
| <code>git merge rama</code> | Es el merge normal que puede crear nuevos commits | De una rama develop a feature. Para resolver los conflictos desde nuestra rama feature ya que nos da igual si hay nuevos commits |

Usos de merge

- Bajar (o actualizar) una rama desde github sin que hayamos hecho ningún commit nuevo en esa rama desde la última vez que la bajamos.

```
git fetch --prune
git switch develop
git merge --ff-only origin/develop
```

```
git fetch --prune
git switch release
git merge --ff-only origin/release
```

```
git fetch --prune
git switch master
git merge --ff-only origin/master
```

- Mergear, develop en release o release en master (suponiendo que están actualizadas ambas ramas porque las hemos actualizado con las órdenes anteriores) y que no hemos hecho commits directamente ni en release ni en master.

```
git switch release  
git merge --ff-only develop
```

```
git switch master  
git merge --ff-only release
```

- Mergear (rebase) los cambios de origin/develop en nuestra develop si hay commits distintos en develop y origin/develop.

```
git fetch --prune  
git switch develop  
git rebase origin/develop
```

- Merge en develop de nuestra rama **con un solo commit** con `--squash`.

```
git switch develop  
git merge --squash raquel  
git commit -am "feat(#42):pantalla de Login"
```

- Mergear los cambios de develop en nuestra rama para ir solucionado los conflictos poco a poco (Antes hemos actualizado develop desde GitHub).

```
git switch raquel  
git merge develop
```

DESPLIEGUE BÁSICO EN SERVIDOR

Comandos

Gestión de imágenes

| Comando | Explicación |
|--------------------------|--------------------------------------|
| <i>Docker image pull</i> | Bajar una imagen de dockerhub |
| <i>Docker image rm</i> | Borrar una imagen en local |
| <i>Docker image ls</i> | Ver las imágenes que hay en local |
| <i>Docker image save</i> | Exportar una imagen a un fichero |
| <i>Docker image load</i> | Importar una imagen desde un fichero |

Comentado [RR1]:

Gestión de contenedores

| Comando | Explicación |
|-------------------------------|--|
| <i>Docker container run</i> | Crear un nuevo contenedor y ejecutarlo a partir de una imagen |
| <i>Docker container start</i> | Iniciar un contenedor que está parado |
| <i>Docker container stop</i> | Parar un contenedor |
| <i>Docker container rm</i> | Borrar un contenedor que está parado |
| <i>Docker container ls</i> | Ver los contenedores que hay |
| <i>Docker container exec</i> | Ejecutar una orden en el contenedor |
| <i>Docker container stats</i> | Mostrar uso de CPU, memoria, etc. De los contenedores |
| <i>Docker container logs</i> | Muestra lo que sale por pantalla. Es decir, lo que se llama en Linux stdout y stderr. Normalmente los contenedores van sacando por pantalla lo que hacen |

Gestión de redes

| Comando | Explicación |
|----------------------------------|-------------------------------------|
| <i>Docker network create</i> | Crea una red |
| <i>Docker network rm</i> | Borrar una red |
| <i>Docker network ls</i> | Ver las redes que hay |
| <i>Docker network connect</i> | Conecta un contenedor a una red |
| <i>Docker network disconnect</i> | Desconecta un contenedor de una red |