Alejandro Martínez Palanques

Florida universitaria  Despliegue aplicaciones web

eXÁMEN DESPLIEGUE

Creamos el repositorio en git:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Clonamos el repositorio en local en mi carpeta llamada “Examen despliegue aplicaciones web”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

EJERCICIO 1

1. Creamos README y lo subimos como primer commit:

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

1. Al “Commitear” sobre origni nos generará master y hacemos el push para subirlo a remoto:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

C / D) Se crea un fichero private.txt y una carpeta private (dentro de la carpeta he de añadir cualquier cosa si no sourcetree no lo detecta)

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Ahora los ignoramos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

E) Nos añade en el fichero gitignore la tanto la carpeta como el txt:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

F) Añadimos el fichero 1.txtInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora comiteamos, pusheamos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

G / H) Una vez pusheado a master le añadimos el tag:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

EJERCICIO 2)

1. Creamos una rama llamada v.02:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Creamos un fichero donde conste la fecha y hora del examen sobre la rama v0.2:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Comiteamos y pusheamos los cambios para que suba la rama v0.2 al repo remoto y luego poder hacer el mergeo sobre master:

Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamente

1. Una vez pusheada la rama v0.2 a remoto desde master podremos mergearnos los cambios que hay sobre la rama:

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Ahora desde master nos indica que tenemos un nuevo cambio por pushear, que es el cambio realizado sobre la rama v0.2:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Finalmente pusheamos sobre master.

EJERCICIO 3) Generamos conflicto:

1. Escribimos hola sobre 1.txt en la rama master y comiteamos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Escribimos adios sobre la rama V0.2 y comiteamos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Tenemos ahora 1 commit por pushear en master y otro commit por pushear en v0.2:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Pusheamos el hola de master y no da ningún conflicto:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Pusheamos v0.2 y tampoco hay ningún problema:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora al traernos v0.2 a master:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nos indica que hay conflictos ya que hemos modificado el mismo fichero:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Entramos a mano en el fichero con el visual studio y marcamos que queremos ambos datos:

Captura de pantalla de computadora

Descripción generada automáticamente

Y lo marcamos como resuelto:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

EJERCICIO4)

1. Para listar las ramas con merge: git branch – merged las que no tienen merge git branch –no-merged
2. Creamos un tag v0.2 sobre la rama v.0.2

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Borramos la rama v.0.2, para ello tenemos que situarnos sobre master ya que si no nos quedaríamos en el “limbo” y no daría ningún error.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente