**Laboratori Gestión de versiones 2**

Somos una startup de tecnología. Hemos desarrollado una aplicación que va a revolucionar el mercado de la música. Este aplicativo permite crear drum loops (o pistas de percusión) y enviarlos a diferentes usuarios.

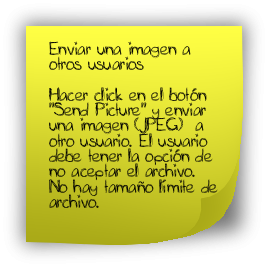
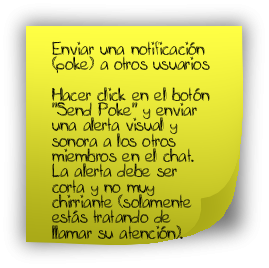
Hace unos días, recibimos una llamada de una persona interesada en invertir en nuestra empresa. Esta persona era el famoso productor de hip-hop Dr Dre (<http://es.wikipedia.org/wiki/Dr._Dre>) que hace pocos días acaba convertirse en el primer billonario del hip-hop por la compra de su servicio Beats por parte de Apple (<http://techcrunch.com/2014/05/09/dr-dre-appears-to-confirm-beatsapple-deal-calls-himself-the-first-billionaire-in-hip-hop/>) y está interesado en invertir en nuevas empresas prometedoras.

Para la primera reunión con él, tenemos que impresionarlo, por lo que se nos requiere implementar dos funciones nuevas:

* Enviar una notificación (poke) a otros usuarios.
* Enviar una imagen a otros usuarios.

Perteneces a un equipo (de 3 personas) que se encarga del desarrollo de la aplicación: Juan, que es el Scrum Master y Pepe y tu que son los developers.

Luego de su reunión de Sprint Planning (Planificación del Sprint), llegan a lo siguiente:



|  |
| --- |
| **Nombre:** HU-123: Enviar una notificación (poke) a otros usuarios. |
| **Prioridad:** 20 |
| **Puntos:** 3 |

|  |
| --- |
| **Nombre:** HU-124: Enviar una imagen a otros usuarios. |
| **Prioridad:** 21 |
| **Puntos:** 4 |

Nuestro CTO hay definido que vamos a utilizar el servidor de versiones llamado GIT. Ya se nos ha registrado y se nos ha dado permiso para poder obtener el código y luego subir nuestros cambios.

**PASO 1. Nos ubicamos en una carpeta proyecto01 que esté controlada por GIT**

Se hace click derecho sobre la carpeta proyecto01 y se escoge Git bash para que se abra una consola. Luego se da el comando:

$ git init

$ git config user.name ”Luis Carpio”

$ git config user.email “[lcarpio@gmail.com](mailto:lcarpio@gmail.com)”

Con ello el directorio proyecto01 se encuentra bajo control de versiones.

**PASO 2. Obtenemos el código de la aplicación del servidor de versiones**

Obtenemos de la página GitHub el código a modificar.

$ git clone <https://github.com/rcheca/GuiaGit.git>

Abrir la consola sobre el proyecto descargado.

**PASO 3. Implementamos la historia “Enviar una notificación (poke) a otros usuarios”.**

Creamos un nuevo método envioNotificacionPoke() en la clase Alumno.

Posteriormente, enviamos nuestros cambios al estado Stage. Para esto ejecutar el siguiente:

$ git status

$ git add .

$ git status

$ git config user.name ”Luis Carpio”

$ git config user.email “[lcarpio@gmail.com](mailto:lcarpio@gmail.com)”

Para realizar un commit y persistir lo cambios, realizar el siguiente comando:

$ git commit –m “HU-123 GrupoXX”

Para verificar el estado de los archivos, hacer:

$ git status

$ git log

**PASO 4. Hacemos push de nuestros cambios al servidor.**

$ git push origin master

**Nota:**

usuario: rcheca

contraseña: Argos1980

**PASO 5. Implementamos la historia de usuario “Enviar una imagen a otros usuarios”.**

Creamos un nuevo método envioImagen() en la clase Alumno.

Llevamos al stage y hacemos un commit para actualizar nuestros cambios localmente.

$ git status

$ git add .

$ git commit –m “HU-124 GrupoXX”

¿Qué problema se ha tenido?

**PASO 6. Hacemos un merge.**

Se obtienen los cambios hechos en el servidor

$ git pull origin master

Si no hay conflictos, la herramienta procede a hacer merge automáticamente.

Caso que si haya conflictos, el merge se tendrá que hacer manual

**PASO7. Creamos una nueva branch de desarrollo**

Para el caso de manejar un nuevo cambio, lo recomendable siempre es crear y manejarlo mediante un nuevo branch.

Para crear un nuevo branch, escribir el siguiente comando:

$ git branch otrobranch

$ git checkout otrobranch

**PASO 8. Creamos un tag para una nueva versión.**

$ git tag –a 1.0 -m “en desarrollo”

**PASO 9. Hacer los cambios necesarios hasta en dos oportunidades**

$ git status

$ git add .

$ git commit –m “HU-124 parte A”

$ git status

$ git add .

$ git commit –m “HU-124 parte B”

**PASO 9. Realizar un merge en el branch Master**

$ git status

$ git checkout master

$ git merge otromaster

**Paso 10. Interpretar el gráfico de versiones**

Haga click derecho sobre el proyecto GuiaGit y escoga Git History. Interprete.

Anexo: Control de versiones distribuido

