**RubyTrick**

**Manejo del Ambiente Controlado**

**Versión [1.1]**

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| [30/08/2014] | [1.0] | Creación del documento | Demián Ardus |
| 31/08/2014 | 1.1 | Revisión de SQA | Santiago Gómez |
|  |  |  |  |
|  |  |  |  |

**Contenido**

[**1.** **DEFINICIÓN DEL AMBIENTE CONTROLADO 3**](http://�)

[**2.** **DESCRIPCIÓN DEL USO DEL AMBIENTE CONTROLADO 3**](http://�)

[**3.** **RESPONSABILIDADES 3**](http://�)

1. **Definición del Ambiente Controlado**

El ambiente controlado se implementara por medio de Git, especificamente GitHub.

1. **Descripción del uso del Ambiente Controlado**

Observaciones:

En la branch principal “master” se encuentra el código revisado y aprobado. Este es el código que va a ser subido o está actualmente en el servidor. Esta branch será entonces la linea base del proyecto.

“origin” es el nombre que hace referencia al repositorio remoto, puede tener cualquier otro nombre pero generalmente se usa este. Para ver los repositorios remotos utilizamos: git remote -v.

La nomenclatura a utilizar para las branches será feature/título donde en la primer parte se especifica la característica de la branch y la segunda el título o nombre de la misma.

Ejemplos: feature/fb\_registration, bugfix/forgot\_my\_password, nonfunctional/css\_refactor.

Pasos para trabajar con el repositorio:

1. Nos paramos en la branch master. (git checkout master).
2. Actualizamos la branch master para estar sincronizados con la principal. (git pull origin master:master).
3. Creamos una nueva branch local a partir de la branch master (git checkout -b feature/NuevaFeature).
4. Programamos la NuevaFeature.
5. Ceamos los commits locales con los cambios programados (git add . y luego git commit -m “Nombre del commit”).
6. Pusheamos nuestro trabajo al servidor remoto creando una branch remota con el nombre indicado (git push origin feature/NuevaFeature:feature/NuevaFeature).
7. En GitHub creamos un Pull Request solicitando que se haga un merge de nuestra Branch a Master.
8. Esperamos la revision de codigo.
9. Una vez revisado y corregido se realiza el Merge.
10. Volver a punto 1 para trabajar en otra nueva funcionalidad.

1. **Responsabilidades**

Cada integrante del grupo del proyecto sera responsable de subir al repositorio el trabajo que le fue asignado, quienes posteriormente revisen este trabajo seran tambien responsables de la correctitud del mismo. En caso de encontrar alguna inconsistencia el Responsable de SCM podra revisar la historia de modificaciones realizadas a la linea base y consultar con los responsables de las mismas para resolver dicha inconsistencia.