***Flujo de trabajo profesional con Pull request***

Antes de introducir el tema de ***Pull request,*** conviene bien dejar clara una noción sobre lo que son los *servidores de desarrollo* (o en ingles, ***staging develop***): básicamente pueden ser definidos como la presentación, de la última versión, de un proyecto en desarrollo (llamese sitio web o aplicación móvil, etc) que está en su última fase de prueba; es decir, es ese espacio previo, para *testear,* lo que queremos *posiblemente* presentar como proyecto final ad portas a ser presentado a un público, usuario, cliente o navegador cualquiera.

Técnicamente los *servidores de desarrollo* tienen su origen en ramas (lo que son) que están justamente por detrás de la rama *master* o *main* (las ramas principales que se suben en el servidor final, la carta de presentación: ***servidor de producción***). Entonces los servidores de desarrollo, que vendrían siendo las ramas que están justamente detrás de la rama *master* o *main,* cuentan también con un servidor que se llama: ***servidor de prueba.***

*Vealo de la siguiente forma:* el ***servidor de producción*** vendría siendo la página web o aplicación movil ya elaborada, que ya superó todas las fases de prueba y que ya está siendo presentada ante usuarios y/o clientes. En cambio, el ***servidor de prueba*** es ese misma página web o aplicación movil en su, *posiblemente,* última versión; la cúal está siendo evaluada por unas últimas fases de prueba para determinar si ya es apta (si se podría considerar o no) como un proyecto finalmente elaborado y exitoso; y poder así ser cargado, posteriormente, en el *servidor de producción*.

*Ahora bien, qué importancia podría tener todo esto en relación a la herramienta de GitHub* ***Pull request?,*** pues que, justo después de que cada desarrollador independientemente haya trabajado en su versión final, según una rama de desarrollo que le haya sido otorgada; todos, sí o sí, deben cargar y fusionar, previa revisión, sus últimos avances *(los que tienen cada uno en sus ramas de desarrollo correspondientes)* primeramente a: la rama del *servidor de desarrollo.* Justo antes de estar todo el código, de todos los desarrolladores con sus respectivas ramas por desarrollador, cargados y fusionados en el servidor de desarrollo: ***se debe revisar*** (como mecanismo de filtro igual, es por eso que es cargado todo el código de todos los desarrolladores, ya fusionado, a esta rama previa a la rama *master: stagind develop*). Ese proceso de revisión que se hace, justo en el momento antes de fusionar en su totalidad (e individualmente) todas las ramas correspondientes a cada desarrollador con la rama *stagind develop,* se le conoce como: ***pull request.***

*Pull request, a modo general, es ese lugar intermedio entre una rama y otra donde se hace un sin número de revisiones (entre lo que exporta una rama e importa otra) antes de llevar a cabo un proceso de fusión entre ambas. Estado intermedio antes de enviar el “merge”.* Esas revisiones se basan o consisten en que otros miembros del equipo puedan ver los cambios que se hicieron, por cada desarrollador, en su rama particular de desarrollo; entonces, hasta que no todos estén de acuerdo en que todos los cambios elaborados era lo esperado, no se hace la fusión entre una rama y la otra (en nuestro ejemplo si una rama de desarrolo en cuestión no gusta, finalmente no podrá hacer un *merge* con *stagind develop*). Ahora bien, en caso que a todos les guste los cambios, éstos son aprobados; y, al aprobarlos, se auto-ejecuta la fusión *merge* en la rama destino; en este caso, siguiendo la línea historica del ejemplo, la rama destino sería *stagind develop.*

Ahora, para hacer la fusión entre *stagind develop* y la rama *master* también se debe pasar por un proceso de ***pull request (herramienta exclusiva de GitHub).***

*un dato: la persona que coordina todas estas revisiones y avala finalmente los cambios y las fuciones entre ramas son los lideres de equipo, muchas veces técnicamente llamados como:* ***DevOps.*** *Un administrador del entorno de desarrollo en su conjunto y que le facilita el trabajo (y lo hace más eficiente) a los desarrolladores.*