

Plan Formativo Full Stack Python	
Módulo	Fundamentos del Desarrollo Web
Tema	Git - GitHub
Nivel de Dificultad	Alta
Ejecución	Grupal (3 ó 4 personas)
Duración	60 Minutos
Código Ejercicio	E.1.6.G1
Intención del aprendizaje o aprendizaje esperado:	
<ul style="list-style-type: none"> <li>Definición de lista de tareas de desarrollo (Backlog) para un sitio web.</li> <li>Coordinación de grupo de desarrollo para logro de objetivos.</li> <li>Adopción de esquema de ramas Git para el flujo de desarrollo del grupo de trabajo.</li> <li>Utilización de ramas Git de tareas individuales y otra de desarrollo convergente.</li> <li>Revisión de desarrollos en rama de pruebas y versionamiento a rama master.</li> </ul>	
Planteamiento del Problema:	
<p>Continuaremos con el sitio de comercio electrónico de ejercicios anteriores, tomando su último estado luego de desarrollar el ejercicio E.1.5.G1.</p> <p>En este caso completaremos tareas simples de trabajo concurrente entre los distintos desarrolladores del proyecto como ocurre en un proyecto real. Para esto estableceremos los siguientes desarrolladores de los participantes: <b>Desarrollador 1, Desarrollador 2, Desarrollador 3, Desarrollador 4, Revisor/Despliegue:</b></p> <p>Existen muchas formas de coordinación en relación a estructura de ramas para un proyecto. Esto dependerá de la comodidad para el grupo de desarrollo y la complejidad del proyecto. En este caso el grupo de desarrollo del proyecto ha decidido trabajar con las siguientes ramas para el proyecto:</p> <ol style="list-style-type: none"> <li>1. master: Última versión aprobada del proyecto y disponible para pasar a producción.</li> <li>2. test: Versión desplegable en producción para pruebas.</li> <li>3. desarrollo: Rama convergente de todos los desarrolladores.</li> <li>4. tarea_xxx: Rama de desarrollo de una tarea con identificador xxx en la lista de pendientes del proyecto.</li> </ol> <p>El flujo de trabajo acordado en este caso es:</p> <ol style="list-style-type: none"> <li>1. Cada desarrollador confecciona nuevas funcionalidades creando ramas <b>tarea_xxx</b> desde rama <b>desarrollo</b>. Para esto será de especial importancia contar con la versión</li> </ol>	

- más actualizada de la rama **desarrollo**, realizando **pull** de ésta antes de crear la nueva rama.
2. Los desarrolladores trabajan localmente en sus ramas **tarea\_xxx**, realizando **commit** hasta tener la funcionalidad respectiva desarrollada y funcional. Esta tarea puede haberse logrado luego de múltiples commits locales. Cada **commit** debe contener una lista clara de las funcionalidades agragadas al código.
  3. Al momento de completar la **tarea\_xxx**, el desarrollador efectuará un **pull** de la rama desarrollo (Que puede contener ya nuevas actualizaciones provenientes de otros desarrolladores). De ser así, deberá realizar en la rama desarrollo un **merge** de la **tarea\_xxx** correspondiente a la funcionalidad que ha desarrollado. En esta etapa deberá resolver posibles conflictos de código que aparezcan por desarrollo paralelo de otros desarrolladores. También deberá realizar push de la rama **tarea\_xxx** al repositorio remoto con el fin de tener un histórico de las tareas completadas.
  4. Una vez realizado el **merge** mencionado, el desarrollador hará **push** a la rama **desarrollo** del repositorio remoto.
  5. El integrante con el rol de **Revisor/Despliegue**, revisará constantemente actualizaciones de la rama **desarrollo** y las pasará a rama **test**. Con esto se congela un estado de desarrollo para probarlo en un servidor, que en este caso será el servidor de pruebas de Visual Studio Code de la máquina de este integrante. Paralelamente los desarrolladores podrán avanzar en paralelo en siguientes desarrollos sin afectar el servidor de pruebas.
  6. Una vez que el código ha sido probado en la rama **test**, el Revisor/Despliegue realizará un **merge** de ésta desde la rama **master** y le asignará un tag con una versión en formato de dos dígitos (Por ejemplo: 0.1, 0.2, 0.3, etc.). Para cada versión el Revisor agregará una entrada de texto en el archivo README.md del proyecto, con las notas de la versión, detallando las nuevas funcionalidades incorporadas respecto de la versión anterior. Estos datos deberán provenir de los mensajes de commit que cada desarrollador incorporó en el desarrollo de su **tarea\_xxx** en particular.

Trabajos a realizar por el equipo de desarrollo:

- a. Reunión breve en la que se definan 7 nuevas funcionalidades simples para el sitio, que impliquen adición de código HTML, CSS y JavaScript. A cada una se les asignará un identificador y pasarán a ser una **tarea\_xxx** (por ejemplo, **tarea\_001**).
- b. Distribuir 2 tareas por cada desarrollador para agregar nuevas funcionalidades al sitio. El Desarrollador 4 estará a cargo de sólo 1 tarea y poseerá a su vez el rol de Revisor/Despliegue.
- c. El Revisor pasará a **test** frecuentemente el código de **desarrollo** y luego de probar su correcto funcionamiento en el servidor Visual Studio Code lo pasará a **master**, desde donde supuestamente puede ser transferido a una rama de producción (No incluida en este ejercicio).
- d. Deben existir al menos 3 versiones emitidas a master en el tiempo de ejecución de este ejercicio y el código final debe incluir todas las tareas del punto a.

**IMPORTANTE:** Todos los integrantes deben mostrar una colaboración equivalente en cantidad de esfuerzo en el resultado obtenido.

#### Recursos Bibliográficos :

[1] Módulo 1 – Contenido 6: “Gestionar el código fuente de un proyecto utilizando GIT para mantener un repositorio de versiones”