

# REPOSITORIO

Software Innova,  
Guía Básica de Uso



Cofinanciado por el  
programa Erasmus+  
de la Unión Europea

## **Repositorio Software Innova, guía básica de uso**

Resultado del proyecto ACACIA (561754-EPP-1-2015-1-CO-EPPKA2-CBHE-JP) cofinanciado por el programa Erasmus+ ACACIA: Centros de Cooperación para el Fomento, Fortalecimiento y Transferencia de Buenas Prácticas que Apoyan, Cultivan, Adaptan, Comunican, Innovan y Acogen a la comunidad universitaria.

**Autor:** Ricardo José Guevara Aragón

Editor: Fernando Octavio Peña

Diseño gráfico y diseño editorial accesible: Pedro Enrique Espitia Zambrano



Esta obra se distribuye bajo Licencia Creative Commons Atribución No Comercial Sin Derivar 4.0 Internacional

## **Agradecimientos**

Esta obra ha sido posible gracias al compromiso de todos los miembros del consorcio del proyecto ACACIA (Centros de Cooperación para el Fomento, Fortalecimiento y Transferencia de Buenas Prácticas que Apoyan, Cultivan, Adaptan, Comunican, Innovan y Acogen a la comunidad universitaria.) y a la cofinanciación recibida por parte de la Comisión Europea a través del Programa Erasmus+.

El consorcio del proyecto ACACIA está conformado por tres universidades europeas y once de América Latina: Universidad Distrital Francisco José de Caldas (UDFJC), Universidad Nacional de Educación a Distancia (UNED), Universidade Nova de Lisboa (UNINOVA), Pontificia Universidad Católica de Valparaíso (PUCV), Universidad Federal do Oeste do Pará  
Crear presentaciones atractivas: Técnicas y comunicación 6 (UFOPA), Universidad Nacional Mayor de San Marcos (UNMSM), Universidad de Antofagasta (UA), Universidad de las Regiones Autónomas de la Costa Caribe Nicaragüense (URACCAN), Universidad Pedagógica Nacional (UPN - Colombia), Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP), Universitatea "1 Decembrie 1918" Alba Iulia (UAB), Corporación Universitaria Iberoamericana (CUI), Universidad Nacional Autónoma de Nicaragua, León (UNAN – León), Universidad Continental (UC), INCI: Instituto Nacional de Ciegos de Colombia, INSOR: Instituto Nacional de Sordos de Colombia, Fundación Sidar - Acceso Universal (de ámbito iberoamericano).

## Tabla de contenido

Página principal y registro .....	3
Panel de control (o “Dashboard”) .....	4
Repositorios .....	6
Creación de un repositorio .....	6
Primera vista de un nuevo repositorio .....	8
Agregar ficheros y commits .....	10
Creación de peticiones pull e incidencias .....	12
Peticiones pull .....	12
Incidencias .....	15
Apéndices .....	16
Migraciones .....	16
Forks .....	17
Lanzamientos .....	17
Glosario .....	19
Webgrafía .....	20

## Página principal y registro

Esta es la primera página que se mostrará al acceder al sitio<sup>1</sup>. Contiene un mensaje de bienvenida y las opciones principales: Inicio de sesión, Registro de un usuario nuevo, Explorar repositorios<sup>2</sup>.

Figura#1



Si es la primera vez que se visita el sitio, es necesaria la creación de un usuario. Seleccionando la opción “Registro”, se mostrará un formulario similar al de la *Figura#2*. En dicho formulario se debe ingresar el nombre de usuario, contraseña (este campo se ingresa dos veces por seguridad) y la dirección de correo.

Una vez ingresados los datos correspondientes, y después de confirmar que son correctos, se puede dar clic en “Registrar cuenta”. Esto creará un nuevo usuario e iniciará sesión con él.

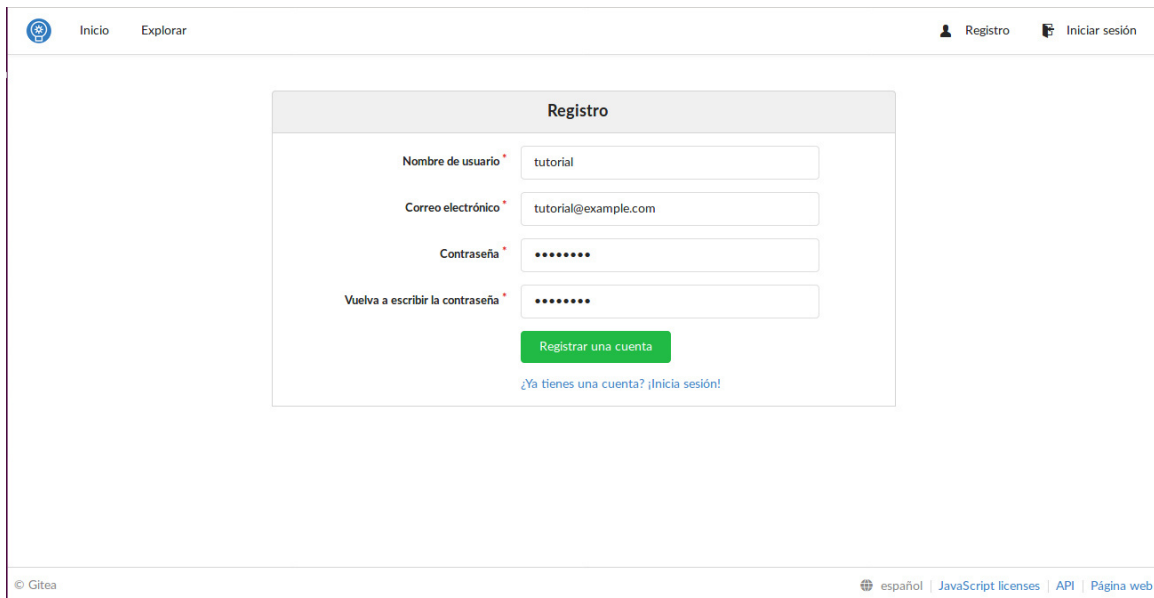
Si al ingresar al sitio, ya se ha creado una cuenta previamente, se debe seleccionar la opción “Iniciar sesión” en lugar de “Registro”.

---

<sup>1</sup> Si el sitio aparece en un lenguaje distinto al que desea, en la parte inferior puede seleccionar el lenguaje de su preferencia.

<sup>2</sup> En el caso particular de Innova Repositories, la exploración de repositorios está permitida solamente a usuarios registrados.

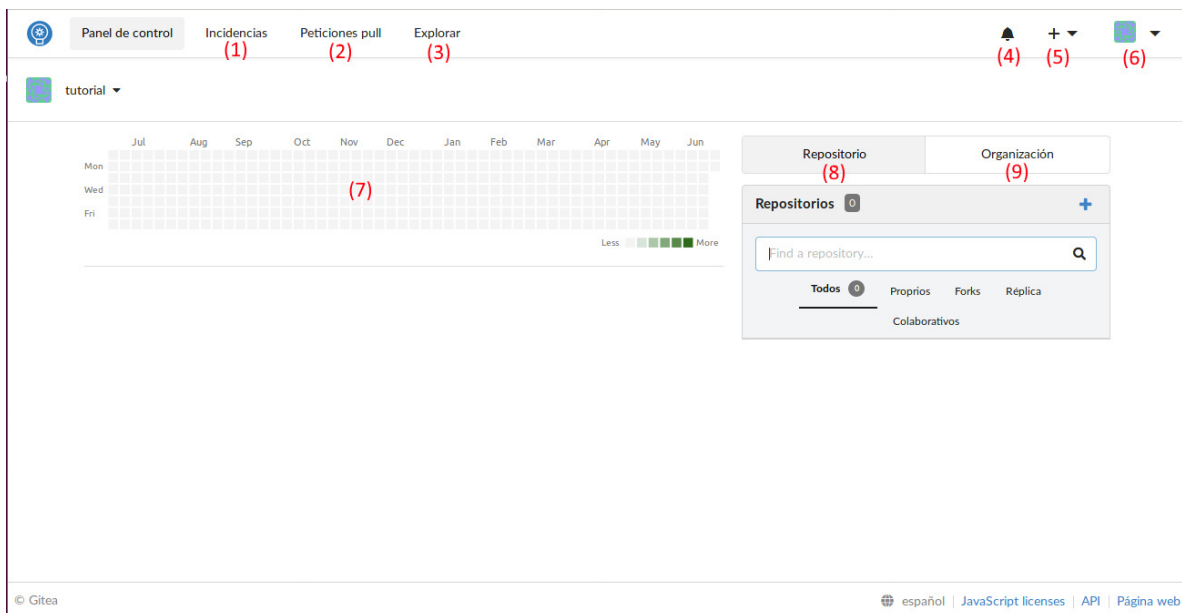
Figura#2



## Panel de control (o “Dashboard”)

El panel de control es el lugar donde la información más relevante referente a una cuenta es mostrada. En la *Figura#3* se puede observar cómo está compuesto.

Figura#3



A continuación, una breve explicación de cada componente numerado en la *Figura#3*<sup>3</sup>:

- 1) **Incidencias:** En esta sección se muestran las incidencias abiertas relacionadas a los repositorios del usuario o a los repositorios en los que el usuario figura como colaborador.
- 2) **Peticiones pull:** Esta sección muestra todas las peticiones pull abiertas relacionadas a los repositorios del usuario y a los repositorios en los que el usuario figura como colaborador.
- 3) **Explorar:** A través de esta opción se puede visualizar de forma general o realizar búsquedas relacionadas a los repositorios públicos, los usuarios del sitio y las organizaciones.
- 4) **Pestaña de notificaciones:** En esta pestaña se mostrarán las notificaciones relevantes como mensajes recibidos, modificaciones hechas, aprobaciones de colaboraciones, comentarios en los repositorios del usuario o en repositorios que el usuario siga, etc.
- 5) **Pestaña de creación:** Es mediante esta pestaña que se accede al formulario de creación de un repositorio o de una migración.
- 6) **Pestaña de perfil:** En esta pestaña están las configuraciones personales del usuario<sup>4</sup>.
- 7) **HeatMap (En español “Mapa de calor”):** Este calendario muestra los días en los que el usuario ha tenido actividad en el sitio y mediante la tonalidad del color revela que días tuvieron más actividad (Entre más oscura la tonalidad, más actividad). En la parte inferior del HeatMap se muestra un historial de las actividades realizadas por el usuario, sin embargo, como la *Figura#3* muestra un usuario recién creado, todavía no hay actividades que mostrar.
- 8) **Visor de repositorios:** En este visor se muestra un listado de todos los repositorios de los que el usuario forma parte, ya sea como propietarios o colaborador. Puede seleccionarse un filtro o mostrar todos los repositorios a la vez.
- 9) **Visor de organizaciones:** En este visor se muestra un listado de las organizaciones a las que el usuario pertenece<sup>5</sup>.

---

<sup>3</sup> No importa si de momento no se tiene una idea clara de lo que las incidencias, peticiones pull o migraciones son, en secciones posterior se abarcan a mayor detalle. También, se referencian en el glosario.

<sup>4</sup> Esta configuración está fuera del ámbito de esta guía por lo cual no será explorada.

<sup>5</sup> En el caso particular de Innova Repositories, la creación de organizaciones no puede ser realizada por un usuario promedio por lo cual esta guía no abarcará ese tema a mayor detalle.

# Repositorios

Los repositorios permiten el almacenamiento de un proyecto preexistente o uno nuevo en la plataforma, de este modo, el proyecto puede beneficiarse de las capacidades de colaboración o simplemente actuar como un respaldo centralizado del código.

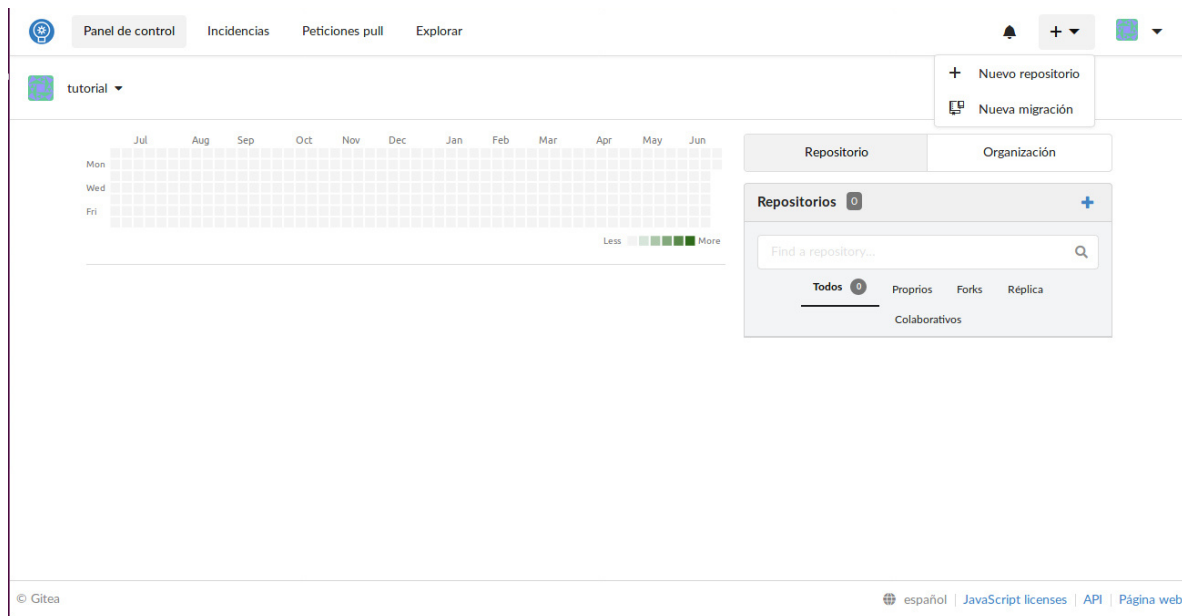
Dentro de un repositorio puede haber muchas ramas y dentro de dichas ramas habrán commits. Cada rama puede ser dedicada a una característica distinta que se planea para la aplicación que se está construyendo o puede utilizarse para resolver problemas o experimentar sin tener que alterar el código principal. Con este enfoque, múltiples personas pueden trabajar en el mismo proyecto, pero no necesariamente estar desarrollando la misma característica. Incluso, si se necesitara, se podría mantener el desarrollo de múltiples versiones de la misma aplicación en paralelo sin mayores complicaciones. Un repositorio puede pasar a ser la dependencia de otros repositorios de forma que puedan nacer proyectos cada vez más grandes.

Tal es la ventaja de un sistema de control de versiones que esta centralizado en un servidor y todo empieza con la creación de un repositorio.

## Creación de un repositorio

Para acceder al formulario de creación de un repositorio, se debe seleccionar la pestaña de creación y luego “Nuevo repositorio” (Figura#4).

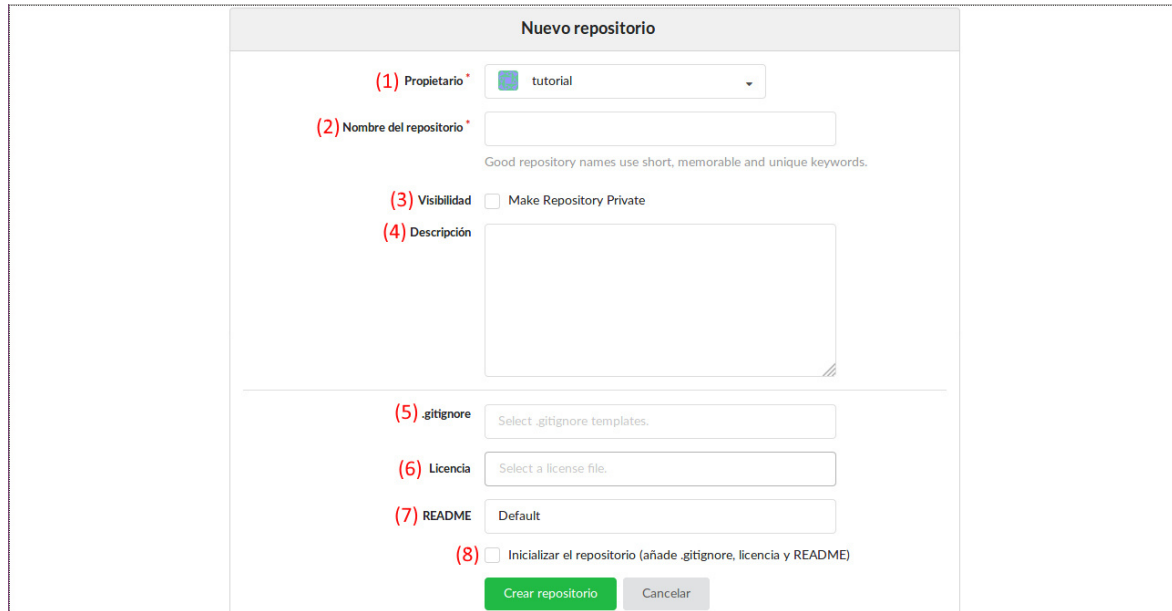
Figura#4





Una vez seleccionada la opción, deberá mostrarse una página similar a la de la *Figura#5*.

*Figura#5*



A continuación, una breve explicación de cada componente en el formulario de la *Figura#5*:

- 1) **Propietario:** Este es el usuario con el que el repositorio está siendo creado y quien administrará dicho repositorio. La razón por la que existe este campo es porque un usuario puede publicar repositorios a nombre propio o a nombre de una organización a la que pertenezca.
- 2) **Nombre del repositorio:** Este es el nombre con el cual el repositorio va a ser identificado; en la pestaña “Explorar” y en búsquedas es este nombre el que aparecerá.
- 3) **Visibilidad:** Los repositorios pueden ser públicos o privados. Si se marca esta opción, el repositorio que está siendo creado será privado; Solamente el propietario y aquellos a quien el propietario designe pueden ver un repositorio privado.
- 4) **Descripción:** Campo opcional. Esta es una descripción relacionada al proyecto, puede ser relacionada a las tecnologías utilizadas, los objetivos del proyecto, etc. Es importante notar que la descripción de un repositorio no es lo mismo que el fichero README.
- 5) **Gitignore:** Campo opcional. “.gitignore” es un fichero que utiliza git para saber que debe ignorar al realizar el seguimiento del proyecto. La plataforma ya incluye muchas configuraciones comúnmente utilizadas. Una vez creado este fichero, puede ser modificado a voluntad para suplir las necesidades específicas que pueda tener el proyecto.
- 6) **Licencia:** Campo opcional. Esta es la licencia que aplica a la utilización del código o productos del repositorio. Si no se está seguro sobre qué escoger o se piensa decidir en un futuro, se puede seleccionar como “Unlicense” para omitir la inclusión de una licencia.

- 7) **README:** Campo opcional. El fichero “README” o “README.md” se utiliza para generar la documentación principal del repositorio; acá pueden incluirse instrucciones de compilación, créditos de autoría, documentación de uso, reglas para los posibles colaboradores, una descripción mas extensa que la que se utilizó en el campo “Descripción”, etc.
- 8) **Inicializar repositorio:** Si este campo no se marca, aunque se hayan llenado los tres campos anteriores ninguno de esos ficheros será incluido al crear el repositorio.

Una vez rellenados los campos obligatorios y, si así se desea, los campos opcionales, basta con dar clic en el botón “Crear repositorio” para finalizar el proceso de creación.

En la *Figura#6* se muestra un ejemplo del formulario completo.

*Figura#6*

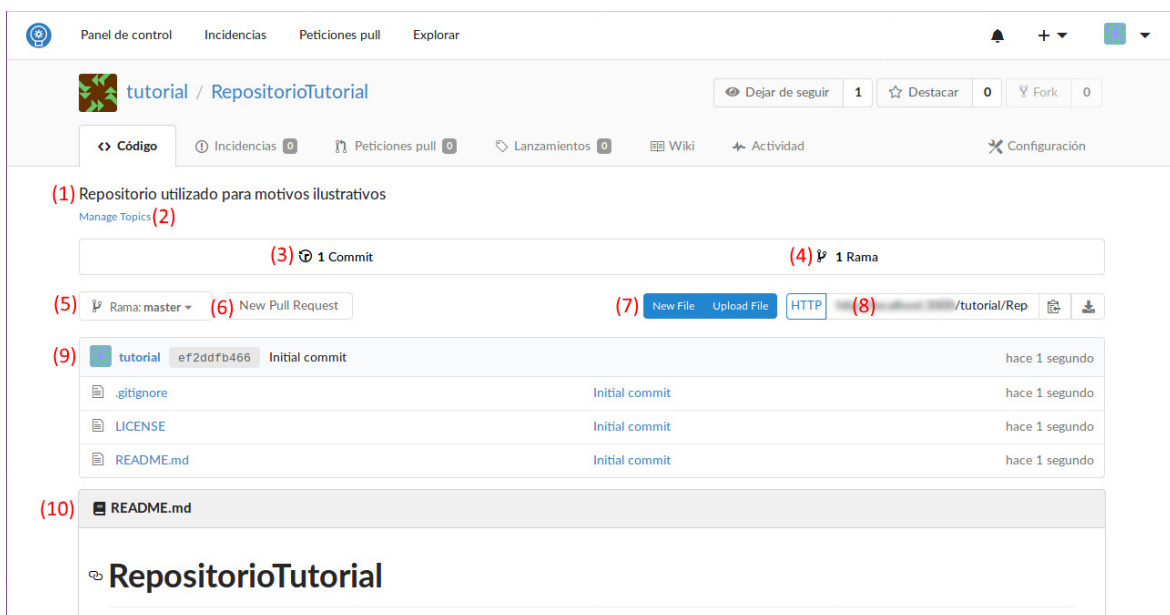
The image shows a web form titled "Nuevo repositorio". The form contains the following elements:

- Propietario:** A dropdown menu with the value "tutorial".
- Nombre del repositorio:** A text input field containing "RepositorioTutorial". Below it is a small note: "Good repository names use short, memorable and unique keywords."
- Visibilidad:** A checkbox labeled "Make Repository Private" which is currently unchecked.
- Descripción:** A text area containing the text "Repositorio utilizado para motivos ilustrativos".
- .gitignore:** A dropdown menu with the value "Node".
- Licencia:** A dropdown menu with the value "Unlicense".
- README:** A dropdown menu with the value "Default".
- Checkboxes:** A checked checkbox labeled "Inicializar el repositorio (añade .gitignore, licencia y README)".
- Buttons:** A green button labeled "Crear repositorio" and a grey button labeled "Cancelar".

### Primera vista de un nuevo repositorio

Siempre que se crea un repositorio nuevo, se muestra el panel de control del repositorio, específicamente en la pestaña de código (véase *Figura#7*). Posteriormente este panel puede ser accedido desde cualquier punto donde el repositorio aparezca listado.

Figura#7



A continuación, una breve descripción de los componentes numerados en la *Figura#7*<sup>6</sup>:

- 1) **Descripción:** Esta es la misma descripción que se ingresa al crear el repositorio.
- 2) **Tópicos:** Administrar las etiquetas que se le han asignado al repositorio o asignarle nuevas. Estas etiquetas son útiles para presentar resultados más relevantes al hacer búsquedas.
- 3) **Commits:** Muestra el número de commits de git que se han hecho en la rama actual.
- 4) **Ramas:** Muestra la cantidad de ramas de git que existen en el repositorio.
- 5) **Selector de rama:** Muestra la rama que está seleccionada actualmente. Cambiar la selección en esta opción es lo mismo que cambiar de rama. Si se ingresa un nombre de una rama inexistente en este campo, se muestra la opción de crearla.
- 6) **New Pull Request:** Este botón es un acceso directo para crear una petición pull desde la rama en la que estamos trabajando.
- 7) **New File y Upload File:** Botones utilizados para crear un nuevo fichero dentro del directorio actual y subir un fichero al directorio actual, respectivamente.
- 8) **Dirección del repositorio:** Esta es la URL hacia el repositorio. Puede ser utilizada para clonar el repositorio localmente o para establecer este repositorio como el origen remoto de nuestro proyecto local en git. Los botones de la derecha son para copiar la URL y descargar el proyecto, respectivamente.

<sup>6</sup> Lamentablemente, la traducción de algunos idiomas en la plataforma está incompleta. Es por ello que algunos botones y vínculos pueden aparecer en inglés.

- 9) **Explorador de ficheros:** Esta sección es para explorar los ficheros del proyecto, los ficheros mostrados y su estado dependen de la rama y commit seleccionados.
- 10) **README:** Muestra el contenido del fichero README.md. Se puede utilizar como una guía rápida o documentación sucinta.

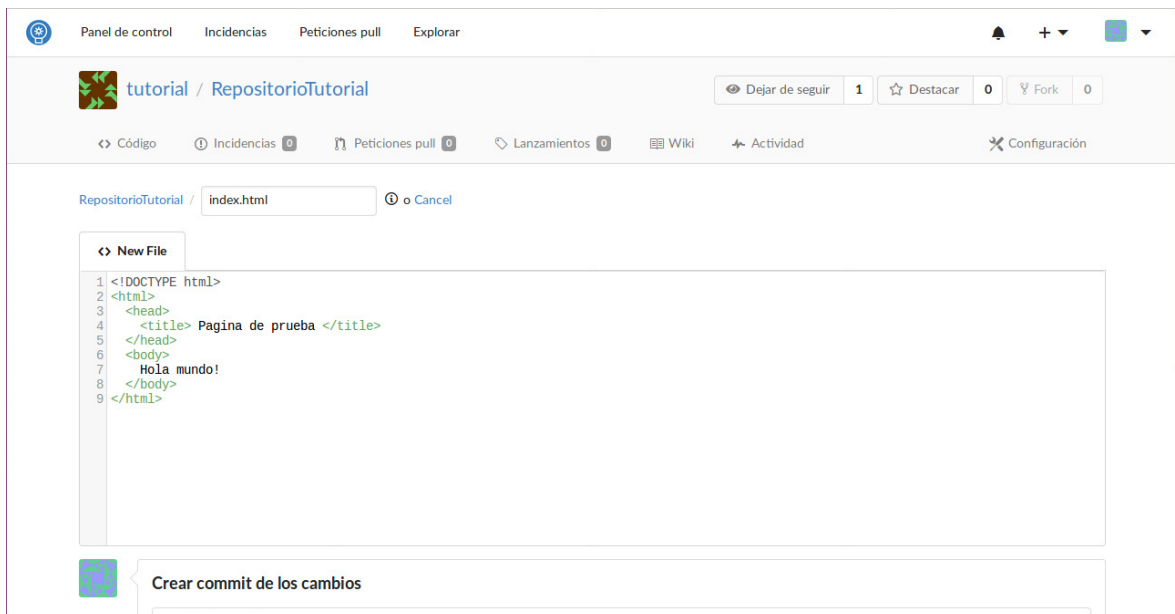
## Agregar ficheros y commits

La plataforma de Innova Repositories pone herramientas a disposición del usuario para la creación y edición de ficheros desde el mismo navegador. Para crear un nuevo fichero, solo basta con seleccionar el botón “New file” y eso redirigirá el navegador hacia un formulario como el de la *Figura#8*.

El formulario está provisto de un campo para nombrar el fichero (con su ruta), un campo para escribir el contenido del fichero y campos de nombre y descripción de un commit (véase *Figura#9*); Esto se debe a que al crear un nuevo fichero o editar uno existente desde la plataforma esta guarda los cambios realizados al proyecto mediante el uso de commits.

Una vez creado un nuevo fichero (y, por consiguiente, un nuevo commit también), se muestra un visor de ficheros que permite ver la sintaxis con color, la versión original, los autores de cada sección (“modo blame”), obtener un enlace permanente al fichero o ver el historial de cambios. Al visualizar los ficheros con el explorador se aprecia que muestran el mensaje del último commit en que fueron afectados al lado (véase *Figura#10*).

*Figura#8*



Figura#9

```
4 <title> Pagina de prueba </title>
5 </head>
6 <body>
7   Hola mundo!
8 </body>
9 </html>
```

**Crear commit de los cambios**

Agregado el fichero 'index.html'

El fichero de prueba 'index.html' se agrego en la raiz del proyecto utilizando el editor en del sitio web.

Hacer commit directamente en la rama `master`.

Crear una nueva rama para este commit y hacer un pull request.

[Crear commit de los cambios](#) [Cancelar](#)

Figura#10

Panel de control Incidencias Peticiones pull Explorar

tutorial / RepositorioTutorial

Dejar de seguir 1 Destacar 0 Fork 0

Código Incidencias 0 Peticiones pull 0 Lanzamientos 0 Wiki Actividad Configuración

Repositorio utilizado para motivos ilustrativos

2 Commits 2 Ramas

Rama: master New Pull Request

New File Upload File HTTP [http://github.com:3000/tutorial/Rep](#)

Nombre del archivo	Descripción	Fecha
tutorial	Agregado el fichero 'index.html'	hace 17 minutos
.gitignore	Initial commit	hace 26 minutos
LICENSE	Initial commit	hace 26 minutos
README.md	Initial commit	hace 26 minutos
index.html	Agregado el fichero 'index.html'	hace 17 minutos

README.md

# RepositorioTutorial

## Creación de peticiones pull e incidencias

Mediante la creación de ficheros y commits, un proyecto puede crecer e incluso eventualmente convertirse en una aplicación madura. Sin embargo, esta forma de actuar todavía no permite la colaboración entre usuarios. Para esto existen herramientas como las peticiones pull y los reportes de incidencias. Múltiples usuarios pueden colaborar en el mismo proyecto, compartir código, discutir la mejor forma de proceder y delegarse tareas sin necesidad de programar reuniones o siquiera tener que estar en la misma habitación.

### Peticiones pull

Las peticiones pull inician discusiones relacionadas a los cambios que se pretenden hacer en un proyecto o a los que se están proponiendo integrar. En cualquier punto del ciclo del desarrollo se pueden abrir peticiones pull y mientras estén abiertas se puede comentar en ellas, llevar cuenta del tiempo que toma llegar a una solución, darles fecha de caducidad (en caso de que la propuesta sea algo que sigue en progreso de desarrollo), estipular dependencias, entre otras cosas.

Luego de una discusión, si existe código que integrar, ya sea agregar código o reemplazar código existente, la última palabra es del propietario del repositorio, quien puede optar por fusionar los cambios o rechazarlos.

Una vez que se decide si una petición pull es rechazada o permitida, esta queda archivada y se marca como “cerrada”. Una petición pull cerrada sigue siendo visible, pero ya no es tomada en cuenta, si se quiere continuar la discusión sobre una característica rechazada o aportar nuevas ideas se debe crear una nueva petición pull.

Para la creación de una petición pull se puede utilizar el botón dentro de la pestaña “Código” (véase *Figura#7*), o se puede utilizar la pestaña “Petición pull” directamente; en esta pestaña se muestran todas las peticiones pull que se encuentran abiertas y también se provee un botón para crear una petición pull propia. El formulario que se deberá completar será uno como el comprendido en las figuras *Figura#11* y *Figura#11.1*.

Figura#11

### New Pull Request

Select the branch to merge into and the branch to pull from.

(1) merge into: tutorial:master ... pull from: tutorial:develop

(2) Agregado el fichero 'ficheroEnDevelop.txt'

(3) Start the title with WIP: to prevent the pull request from being merged accidentally.

(5) Write Preview

(4) Etiquetas Sin etiquetas

(6) Milestone Sin Milestone

(7) Assignees No Assignees

Drop files or click here to upload.

Crear Pull Request

Figura#11.1

Drop files or click here to upload.

Crear Pull Request

1 Commits 30e47fd57d ... 970ec31f3c

Autor	SHA1	Mensaje	Fecha
(8) tutorial	970ec31f3c	Agregado el fichero 'ficheroEnDevelop.txt'	hace 7 minutos

Se han modificado 1 ficheros con 1 adiciones y 0 borrados

(9) +1 -0 ficheroEnDevelop.txt Ver fichero

```
@@ -0,0 +1 @@
1 +Este fichero existe solamente para demostrar la diferencia al navegar entre ramas.
```

© Gitea [español](#) | [JavaScript licenses](#) | [API](#) | [Página web](#)

A continuación, una breve descripción de los componentes enumerados en las figuras *Figura#11* y *Figura#11.1*:

- 1) **Selección de rama de origen y destino:** Si la intención de la petición pull es integrar código a otra rama, es imperativo que la rama de destino se seleccione en la izquierda y la rama de origen en la derecha. También, la rama de la derecha tiene que estar adelantada o ser divergente en comparación a la de la izquierda.
- 2) **Nombre de la petición pull:** Toda petición pull tiene un nombre que la identifica en listas y búsquedas.
- 3) **Botón para marcar la petición como WIP:** WIP (Work In Progress) es algo que solamente se marca en el nombre de la petición pull para dar a entender que la petición es para discutir la dirección de una rama (ya sea para una característica o parche) o para controlar el progreso y no para la implementación inmediata. Al ver esto en el nombre de una petición se debe entender que la implementación todavía no funciona o no es estable.
- 4) **Etiquetas:** Estas etiquetas funcionan similar que las que se les pueden aplicar a los repositorios; ayudan a clasificar las peticiones pull para listarlas u optimizar búsquedas.
- 5) **Descripción:** Esta descripción de la petición pull debe ser bien clara referente a la intención de la petición y, de haber alguno, los conflictos que se pretenden solucionar en la discusión. Esta descripción puede incluir anexos como imágenes o documentos.
- 6) **Milestones:** Estas son metas a mediano y corto plazo referentes al desarrollo de la rama que se quiere integrar con esta petición pull.
- 7) **Assignees:** Las personas asignadas a lidiar con esta petición pull o a resolver las tareas que se puedan discutir en ella.
- 8) **Commits:** El listado de los commits de diferencia entre la rama de destino y la de origen.
- 9) **Diferencias:** Listado de las diferencias entre las dos ramas propuestas.

Una vez creada una petición pull, queda abierta para ser discutida por cualquiera mientras no sea marcada como cerrada o se integre a la rama solicitada (véase *Figura#12*).

*Figura#12*



## Incidencias

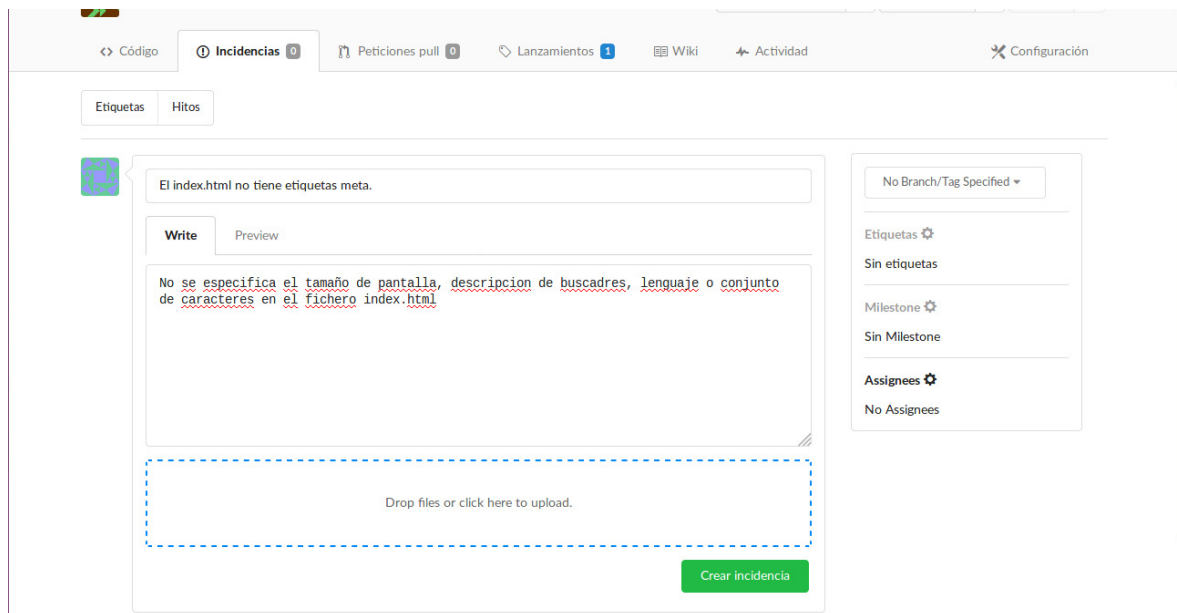
Las incidencias son una gran manera para reportar problemas con una aplicación en desarrollo o sugerencias para mejorar funcionalidades establecidas. La diferencia más grande con respecto a las peticiones pull es que las incidencias usualmente son abiertas por gente que no es colaboradora del proyecto y no sugieren implementaciones de código. Por ello, presentan una gran oportunidad para obtener feedback de terceros.

Otra gran utilidad de las incidencias es la capacidad de utilizarse como un listado de “pendientes”; se puede abrir una incidencia por cada característica pendiente que se piensa implementar y de esa forma llevar cuenta del progreso de la aplicación de una forma pública que permite discusión.

La creación de una incidencia es similar a la de una petición pull, con las diferencias de que no requiere seleccionar ramas (aunque si es posible especificar una rama de origen para la incidencia), no muestra diferencias de commits o cambios en el código.

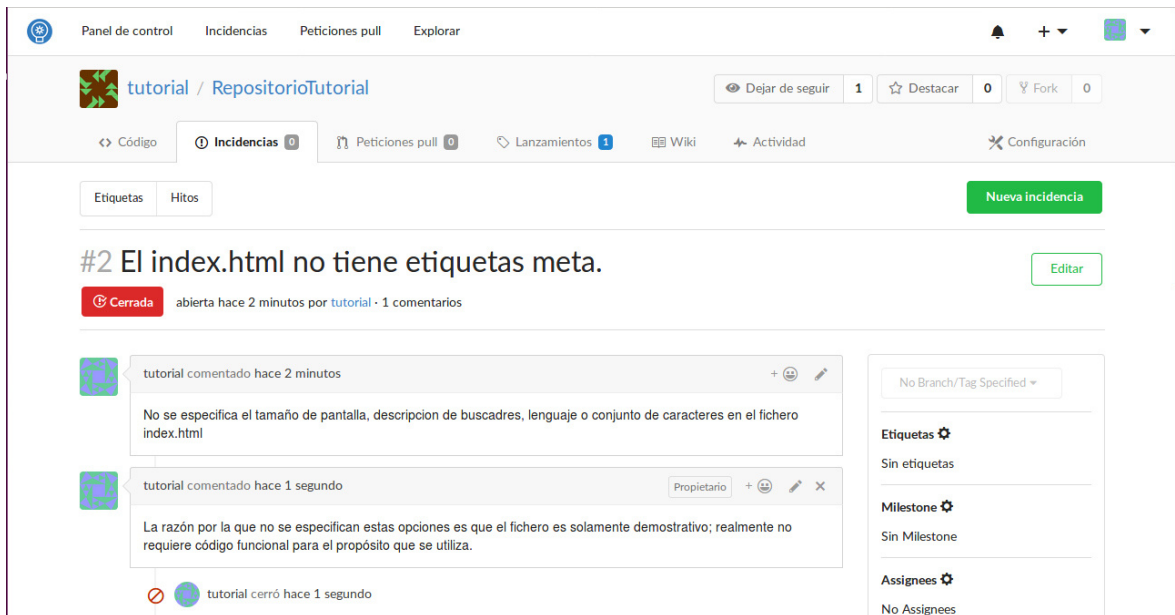
Para crear una incidencia, se utiliza la pestaña “Incidencias”, solo se necesita el nombre de la incidencia y una descripción de esta (véase *Figura#13*). Opcionalmente se pueden incluir anexos (como en las peticiones pull).

*Figura#13*



Una vez creada una incidencia, se puede comentar en ella mientras no sea cerrada por el autor o algún colaborador.

Figura#14

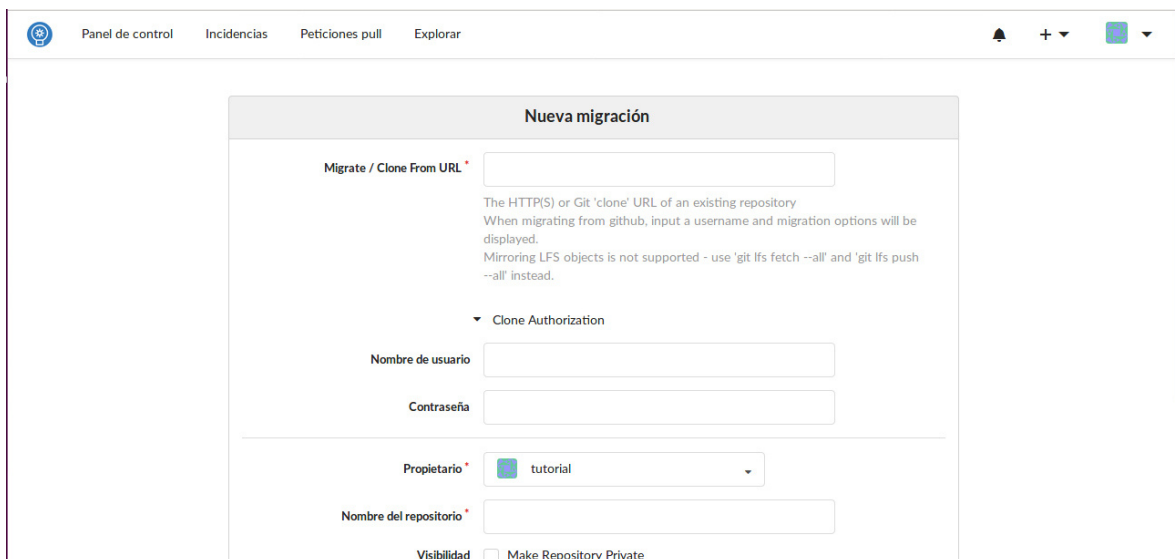


## Apéndices

### Migraciones

Las migraciones son repositorios importados de algún otro servicio (principalmente github). El formulario de creación es ligeramente distinto ya que requiere proveer una URL hacia el repositorio original y las credenciales para el inicio de sesión (véase *Figura#15*) pero una vez importado funciona y se administra de una forma casi idéntica a un repositorio nativo.

Figura#15



## Forks

Un fork es un clon de un repositorio ajeno. Este clon tiene como propietario al usuario que realizó la copia y eso le permite realizar los cambios que desee al repositorio sin afectar al original.

Una característica particular de realizar un fork es que se puede hacer una petición pull hacia el repositorio original para solicitar que los cambios que el nuevo propietario hizo en su copia se implementen. Esta es otra manera en la que un proyecto puede crecer y madurar; el ejemplo más común de esta situación es cuando un usuario realiza un fork de un repositorio para incluirlo como dependencia en un proyecto propio y luego hace cambios para que pueda adaptarse a sus necesidades específicas. Si considera que dichos cambios pueden ser útiles para otras personas puede hacer una petición pull al repositorio original para que sean incluidos.

El botón para realizar un fork está disponible en todos los repositorios mientras no sean propios ya.

## Lanzamientos

Los lanzamientos son versiones estables descargables de una aplicación que esta siendo desarrollada en un repositorio. La distribución de código fuente se generan a partir del commit actual en la rama seleccionada en el formulario de creación (usualmente “master”) y pueden incluir otros tipos de distribuciones como anexo (tar, zip, distribuciones binarias, apks, etc.).

Figura#16

**Nueva Release**  
Releases organize project versions.

0.7-Alpha @ master

Choose an existing tag or create a new tag.

**Título**  
Alfa de nuestro repositorio

**Contenido**  
Esta es la versión alfa de nuestro repositorio de prueba

Drop files or click here to upload.

Mark as Pre-Release  
Mark this release unsuitable for production use.

Si el lanzamiento de una aplicación es marcado como “Pre-release” se especifica que a pesar de ser descargable y utilizable, todavía no es completamente estable o no tiene todas las funciones que se planean incluir.

Figura#17

Panel de control Incidencias Peticiones pull Explorar

tutorial / RepositorioTutorial

Dejar de seguir 1 Destacar 0 Fork 0

Código Incidencias Peticiones pull Lanzamientos Wiki Actividad Configuración

### Lanzamientos Nueva Release

**Pre-lanzamiento**

- 0.7-Alpha**  
8838cd332b

**Alfa de nuestro repositorio (editar)**

tutorial hace 19 minutos 0 commits en master desde esta release

Esta es la versión alfa de nuestro repositorio de prueba

#### Descargas

- [Código Fuente \(ZIP\)](#)
- [Código Fuente \(TAR.GZ\)](#)

© Gitea [español](#) | [JavaScript licenses](#) | [API](#) | [Página web](#)

## Glosario

- **Commit:** En git, un commit es la confirmación para almacenar los cambios realizados como una “instantánea” del proyecto (una copia del estado exacto del proyecto al momento de la confirmación).
- **Feedback:** Retroalimentación.
- **Fork:** Un clon de un repositorio con la persona que hace la copia como propietaria.
- **HTTP:** “HyperText Transfer Protocol” es un protocolo utilizado para la transferencia de datos a través de internet. HTTP es un protocolo sin estado, lo que significa que no guarda de forma inherente un registro de conexiones anteriores.
- **Incidencia:** Reporte de algún mal funcionamiento en el código o sugerencias para mejorarlo de alguna forma.
- **Peticiones pull (Pull Requests):** Es una discusión que puede ser utilizada para proponer nuevas características, revisión de dependencias o solicitar la inclusión o reemplazo de código en un proyecto.
- **Rama:** Una rama es un apuntador hacia un commit de un proyecto; en git siempre existe una rama por defecto llamada “master”.
- **Repositorio:** Lugar donde se guarda algo. En el contexto de Innova Repositories, es la unidad organizacional utilizada para el almacenamiento del código de un proyecto.

## Webgrafía

- Primeros pasos con github. <https://guides.github.com/activities/hello-world/>
- Flujo de un repositorio. <https://guides.github.com/introduction/flow/>
- Dominando incidencias. <https://guides.github.com/features/issues/>
- Haciendo forks. <https://guides.github.com/activities/forking/>
- Ramificaciones en git. <https://git-scm.com/book/es/v1/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>