

# Paradigmas de Programación

## Instrucciones para la primera entrega de prácticas

A continuación se describen los pasos necesarios para realizar la entrega de las **dos primeras prácticas** de la asignatura Paradigmas de Programación (PP). La fecha límite para esta entrega es el viernes **6 de noviembre de 2020** (a cualquier hora de este día).

## Material a entregar

Cada alumno debe subir a su repositorio de PP dos directorios: **p1** y **p2**. Dichos directorios deben contener ÚNICAMENTE código fuente (es decir, ficheros **.ml**). Más concretamente:

- El directorio **p1** debe contener los ficheros **nombre.ml**, **expresiones.ml**, **expresiones2.ml** y **condis.ml**.
- Y el directorio **p2** debe contener los ficheros **curry.ml** y **comp.ml**, y opcionalmente el fichero **ej23.ml**.

## Comprobaciones previas

Asegúrese de que todos los ficheros compilan. Las órdenes de compilación podrán ser las siguientes:

```
ocamlc nombre.ml
ocamlc expresiones.ml
ocamlc expresiones2.mli expresiones2.ml
ocamlc condis.ml

ocamlc curry.mli curry.ml
ocamlc comp.mli comp.ml
ocamlc ej23.mli ej23.ml
```

Como se puede observar, algunas de estas órdenes hacen referencia a ficheros de interfaz (con extensión **.mli**), los cuales se proporcionan junto con estas mismas instrucciones de entrega.

## Método de entrega

La entrega se realizará en el gitlab de la facultad, disponible en <https://git.fic.udc.es>.

Como configuración inicial de gitlab, tenemos que crear una clave pública/privada rsa que git usara para autenticarse a través de ssh contra el servidor:

1. Si ya hemos creado la clave con anterioridad para otros usos, podemos ver la parte pública con:

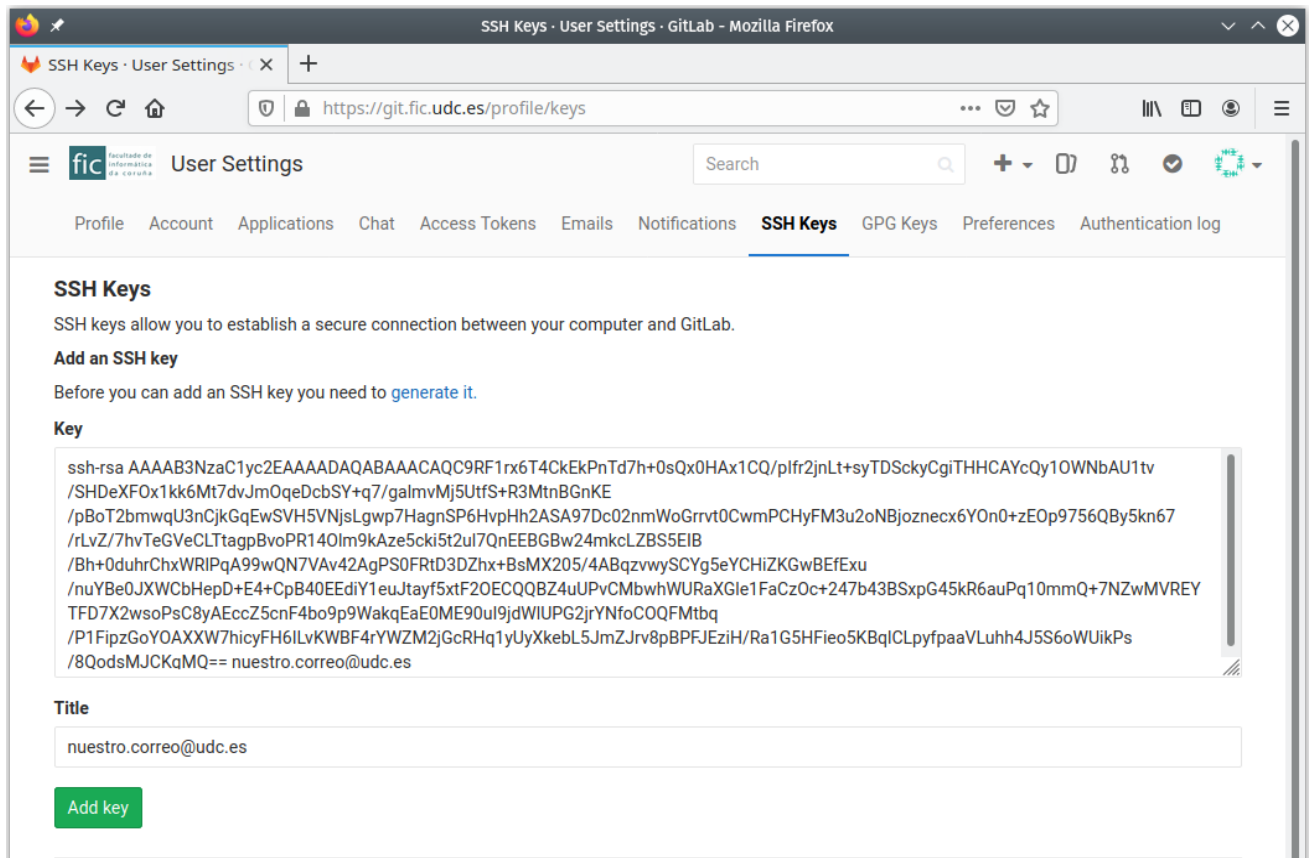
```
cat ~/.ssh/id_rsa.pub
```

Si el comando falla porque el fichero no existe, creamos una clave con:

```
ssh-keygen -t rsa -C "nuestro.correo@udc.es" -b 4096
```

Durante la ejecución, nos pedirá un fichero donde guardar la clave (podemos usar la opción por defecto), y una contraseña para proteger el acceso a esa clave.

2. Abrimos el fichero `~/.ssh/id_rsa.pub`, y copiamos su contenido.
3. Abrimos la url <https://git.fic.udc.es/profile/keys>, copiamos la clave en la ventana, y la añadimos.



Si no tenemos git instalado, lo hacemos usando la herramienta de gestión de paquetes de nuestra distribución. Por ejemplo, en distribuciones basadas en debian (como ubuntu o mint):

```
sudo apt install git
```

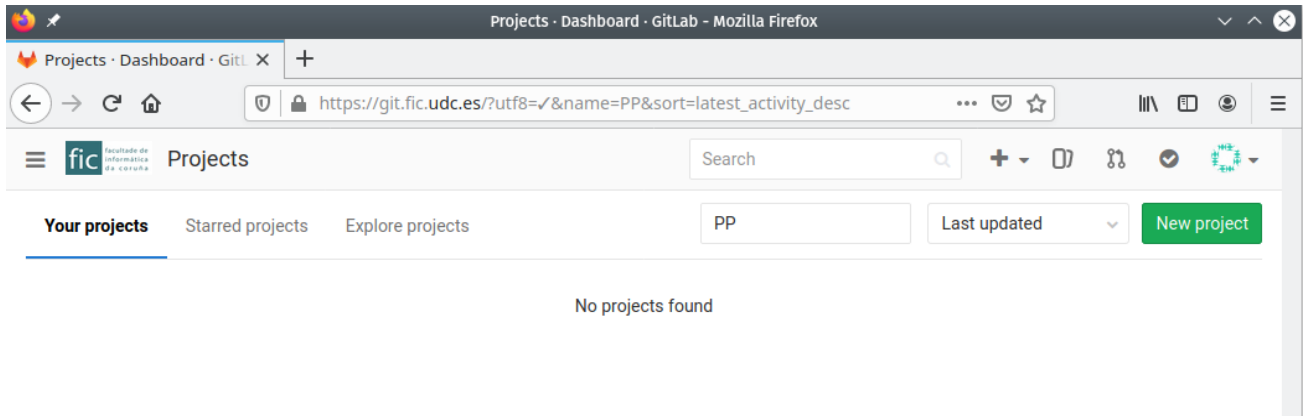
Si nunca hemos usado git, configuramos nuestro nombre y correo electrónico:

```
git config --global user.name "Mi nombre"
git config --global user.email "mi.login@udc.es"
```

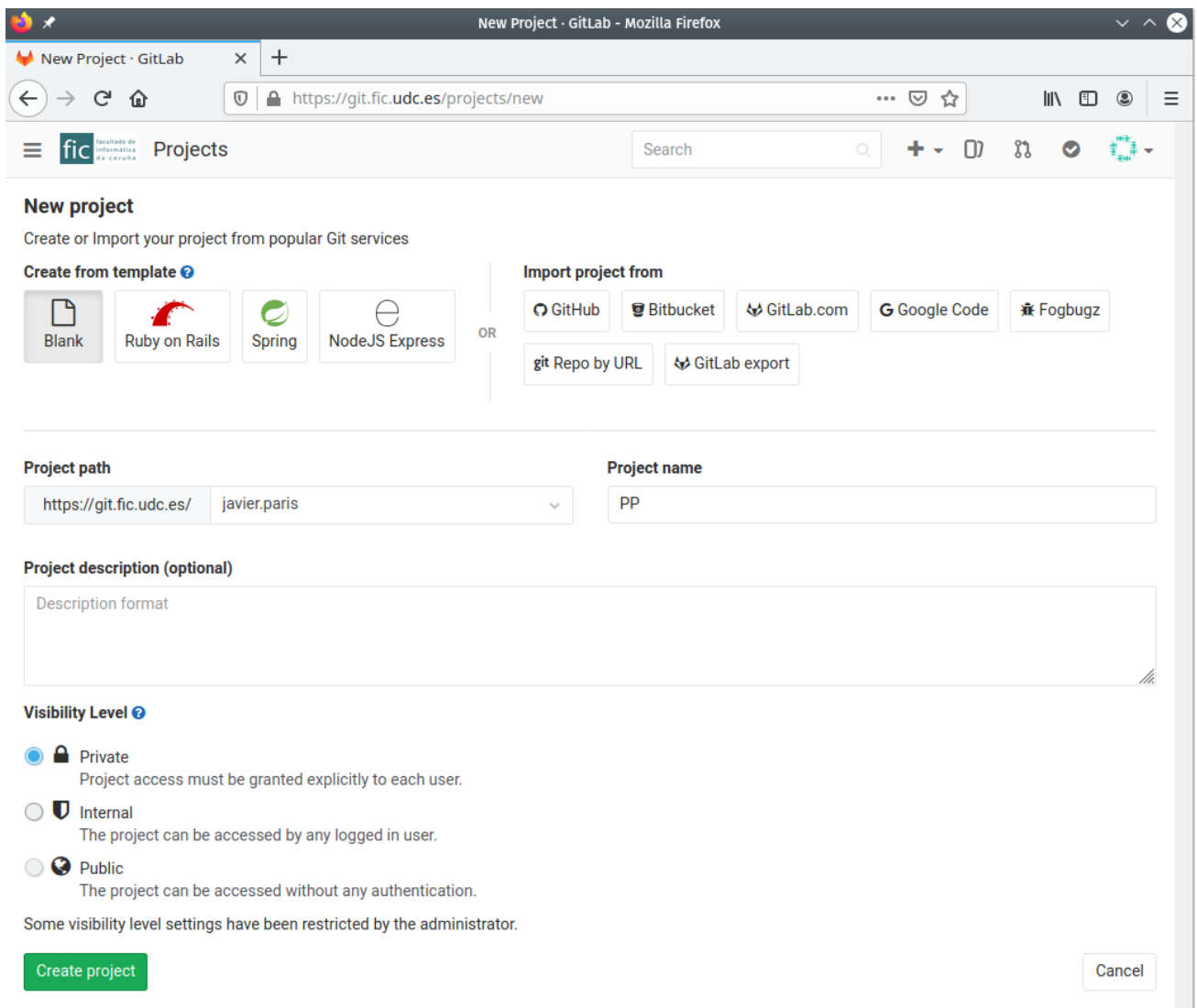
Estos datos se asociarán a cada operación que hagamos contra el repositorio git, tanto en local como en el repositorio guardado en gitlab.

Para la entrega de prácticas, crearemos un repositorio “PP” de la siguiente forma:

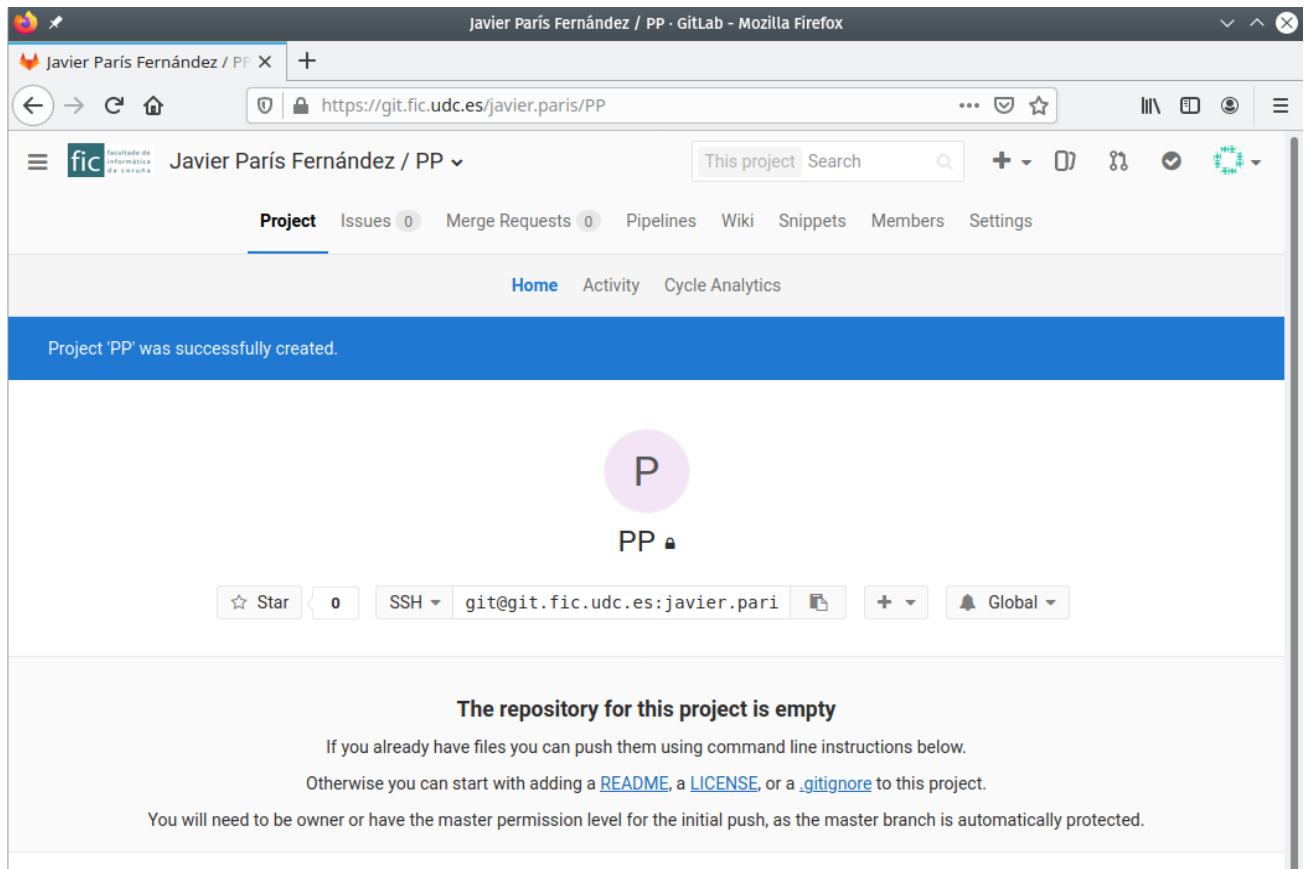
1. Crearemos un proyecto en gitlab seleccionando New Project en <https://git.fic.udc.es/>.



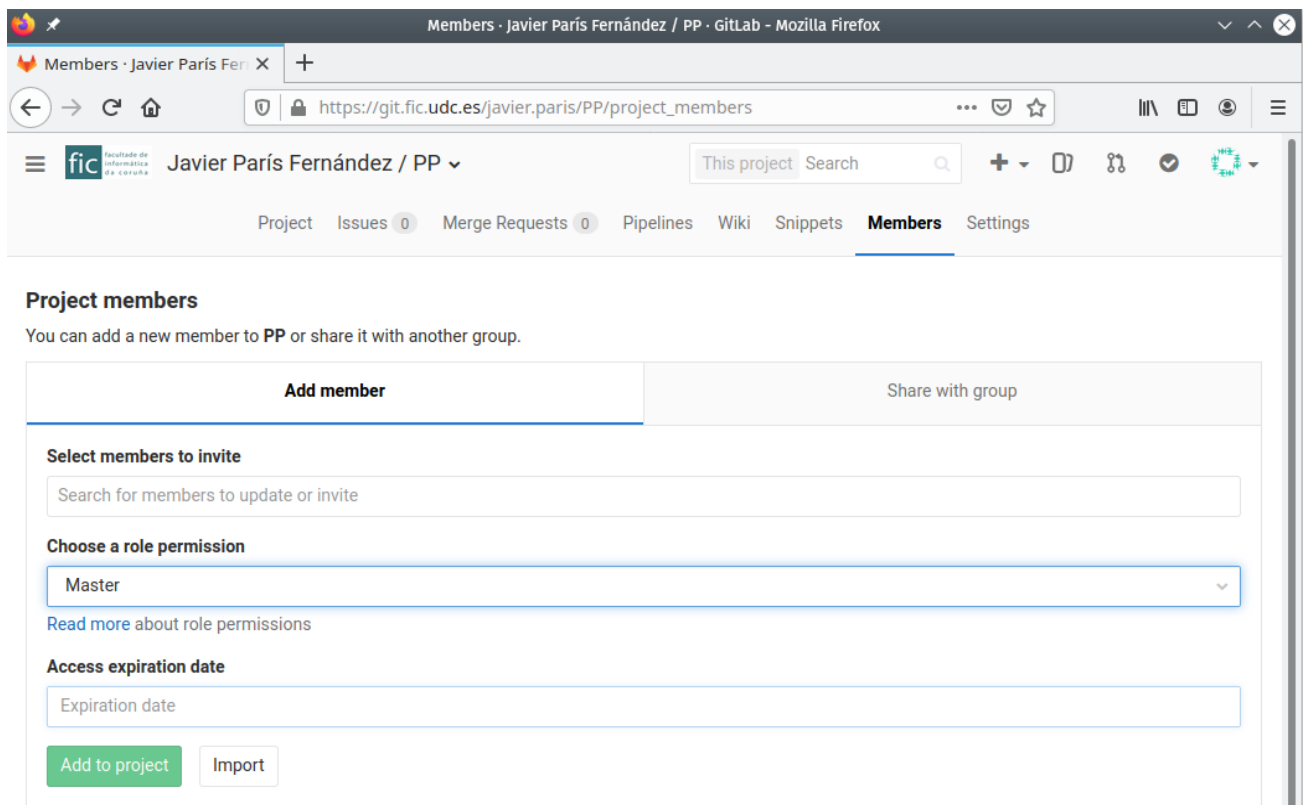
2. En la nueva página, damos nombre “PP” al proyecto, comprobamos que el proyecto es privado, y confirmamos la creación.



3. Una vez creado, nos llevará a la página del proyecto, que inicialmente está vacío. En el cuadro SSH podemos ver la url que usaremos para operar remotamente con él.



4. En la página anterior seleccionamos la pestaña Members, y dentro de ella añadimos a los profesores de la asignatura con rol Master.



5. En nuestra máquina, bajamos el repositorio vacío en el directorio que queramos. La url del proyecto la podemos encontrar en la página gitlab del mismo, en el cuadro SSH.

```
git clone git@git.fic.udc.es:mi.login/PP.git
```

6. Git descargará el proyecto en un directorio con el mismo nombre. Copiamos los directorios y ficheros a entregar en ese directorio. En el terminal, cambiamos al directorio del proyecto:

```
cd PP
```

7. Para indicarle a git que queremos que incluya estos ficheros en el repositorio, usamos para cada uno de ellos la orden:

```
git add fichero.ml
```

Si añadimos un directorio, git añade automáticamente todo su contenido. Es decir, en el caso de esta primera entrega, es suficiente con hacer:

```
git add p1  
git add p2
```

8. Esto no guarda los ficheros en el repositorio, solo le indica a git la lista de operaciones que queremos realizar. Para confirmarlo, tenemos que hacer un commit, que realizará todas estas operaciones contra nuestro repositorio git local:

```
git commit -m "Entrega 1"
```

9. Por último, subiremos estos cambios al repositorio git del gitlab con el comando:

```
git push -u origin master
```

Si realizamos cambios en alguno de los ficheros que queremos incluir en el repositorio de gitlab para la entrega, simplemente repetiremos los 3 últimos pasos (add, commit y push), indicando en el mensaje del commit los cambios que hemos realizado.

**Nota:** Sea muy cuidadoso a la hora de crear los directorios y los ficheros, y respete los nombres indicados. En particular, fíjese que todos estos nombres sólo contienen letras en minúsculas, números y puntos.