

# Hada Práctica 1: Uso de git

---

Departamento de Lenguajes y Sistemas Informáticos Universidad de Alicante

# Objetivos de la práctica.

- Aprender a usar **git** de manera individual y de forma local.
- Aprender a crear repositorios, commits, tags, etc.
- Aprender a crear ramas y trabajar con ellas.
- Realizar nuestro primer programa en C#.

# Uso de git

- En esta primera práctica vamos a hacer uso del sistema de control de versiones **git**.
- Sigue los pasos indicados, **respetar el uso de mayúsculas y minúsculas** así como el **nombre de las carpetas, archivos, clases y métodos** que se te indique que has de crear.
- Al final del documento se indican las condiciones de entrega, los requisitos técnicos que debe cumplir la entrega para ser evaluada y una guía de evaluación de esta práctica.

# Parte I

- Crea una carpeta llamada **hada-p1**
- Dentro de ella crea un archivo C# llamado **p1.cs** el cual tendrá una clase llamada **HadaP1**
- Inicializa el repositorio git y añádele el archivo **p1.cs** sólo con la clase, sin ningún método. Haz el commit correspondiente.
- A continuación añade a esta clase un método como este:

```
public static bool isPalindrome(string s)
```
- Como su nombre indica, dada una cadena "*s*" nos devolverá *cierto* si "*s*" es un palíndromo y *falso* si no lo es. Su implementación debe ser iterativa y deberá tener en cuenta si los caracteres están en mayúsculas o minúsculas.
- Haz el *commit* correspondiente que recoge el añadido de este método.

## Parte II

- Modifica el método anterior para que descarte los espacios en blanco.
- Haz el *commit* correspondiente que recoge esta modificación.
- Etiqueta este commit con el nombre 'nows'.
- Modifica de nuevo el método para que no distinga si un carácter está en mayúsculas o minúsculas, es decir, considerará que 'A' == 'a' , 'B' == 'b', etc...
- Haz el *commit* correspondiente que recoge esta modificación.
- Etiqueta este commit con el nombre 'nocase'.

## Parte III

- Crea una nueva rama llamada `'recursion'` y cámbiate a ella.
- Modifica el método `'isPalindrome'` tal y como está al final de la 'Parte II' para que ahora sea recursivo.
- Haz el *commit* correspondiente que recoge esta modificación.

## Parte IV

- A partir del último commit de la rama 'master', crea ahora una nueva rama llamada 'master2'. Cámbiate a ella.
- Mezcla los cambios de 'recursion' en esta rama.
- A partir del commit llamado 'nows' de la rama 'master', crea ahora una nueva rama llamada 'nows2'. Cámbiate a ella.
- Mezcla también los cambios de 'recursion' en esta rama.

# Entrega.

- La entrega de esta práctica consiste en el directorio de trabajo hada-p1, junto con todo su contenido, comprimido en un fichero llamado hada-p1.tgz.
  - Este archivo lo puedes crear así en el terminal:  
tar cfz hada-p1.tgz hada-p1
- **Lugar y fecha de entrega:** La entrega se realizará en <http://pracdlsi.dlsi.ua.es> en las fechas allí publicadas.
- **No se admitirá ningún otro método de entrega.**



# Requisitos técnicos.

Requisitos que tiene que cumplir este trabajo práctico para ser evaluado (si no se cumple alguno de los requisitos la calificación será **cero**):

- El archivo entregado se llama `hada-p1.tgz` (todo en minúsculas).
- Al descomprimir el archivo `hada-p1.tgz` se crea un directorio de nombre `hada-p1` (todo en minúsculas).
- El directorio `hada-p1` contiene la copia de trabajo y el directorio `.git`.
- Los archivos que hay dentro del directorio `hada-p1` se llaman como se indica en el enunciado (respetando en todo caso el uso de mayúsculas y minúsculas).
- Las etiquetas, nombres de ramas, clases C# y métodos implementados se llaman como se indica en el enunciado (respetando en todo caso el uso de mayúsculas y minúsculas).

# Guía de evaluación.

- Los distintos commits pedidos a lo largo del enunciado supondrán hasta el 30% de la nota.
- La creación correcta de las etiquetas pedidas supondrá hasta el 10% de la nota.
- La creación y trabajo correctos con las ramas pedidas supondrá hasta el 60% de la nota.