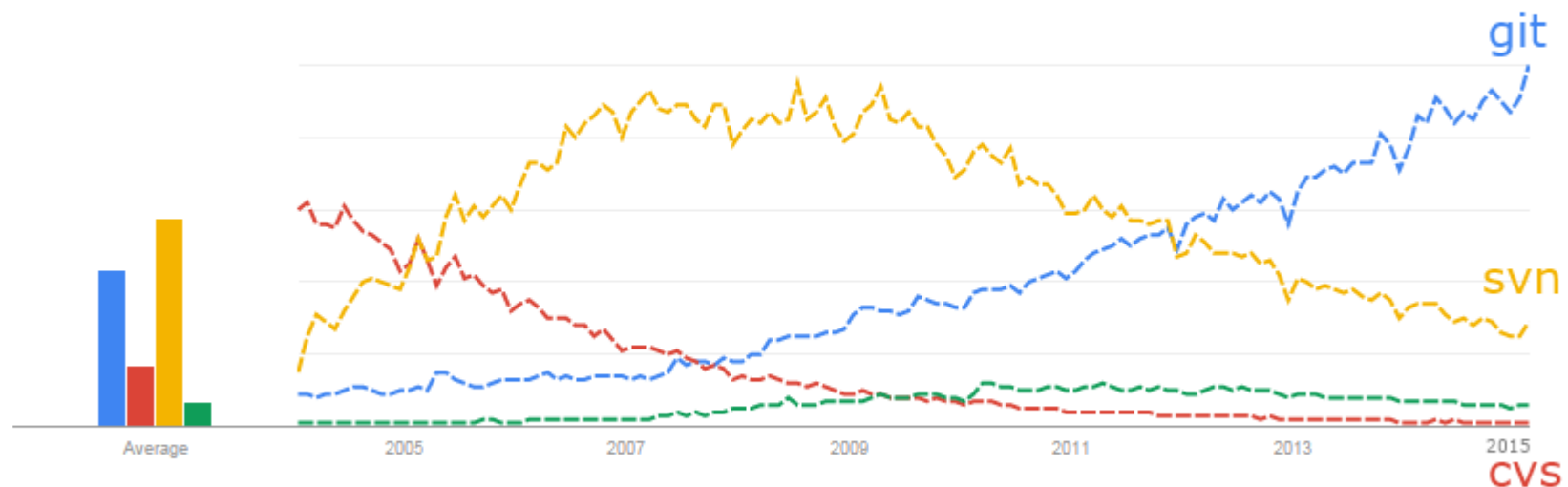


# Introducción a Git

Oscar Delgado  
oscar.delgado@uam.es

# ¿Qué es Git?

- De largo, el sistema de control de versiones distribuido más popular
- Estándar de facto en empresas e instituciones



# ¿Qué es Git?

- Creado en 2005 por Linus Torvalds para manejar el caos del kernel de Linux (> 16M de líneas de código)
- Mucho **más rápido, robusto y estable** que cualquier otro sistema existente.
- Obviamente, *open source*.

# Principales actores



# Conceptos básicos

Repositorio

Rama

Commit

Stage area

Directorio de  
trabajo

# Repositorio

- Contiene la historia y diferentes versiones de cada fichero del proyecto.
- Típicamente, almacena una copia local de un repositorio remoto.
- Operaciones típicas:
  - **Creación** de un nuevo repositorio.
  - **Clonado** de un repositorio existente.

# Creación de un repositorio

- Desde el directorio que contiene el código:

```
# git init
```

- Esto crea el repo solo localmente.

# Clonado de un repositorio

- Inicialización del repo desde la web del proveedor (Github, Gitlab, etc.)
- Clonado del repo vacío a local:
- ```
# git clone http://vega.ii.uam.es/  
[GRUPO] [NUM_PAREJA] /practical.git
```



# Configuración básica

- Configuración básica de vuestros usuarios:

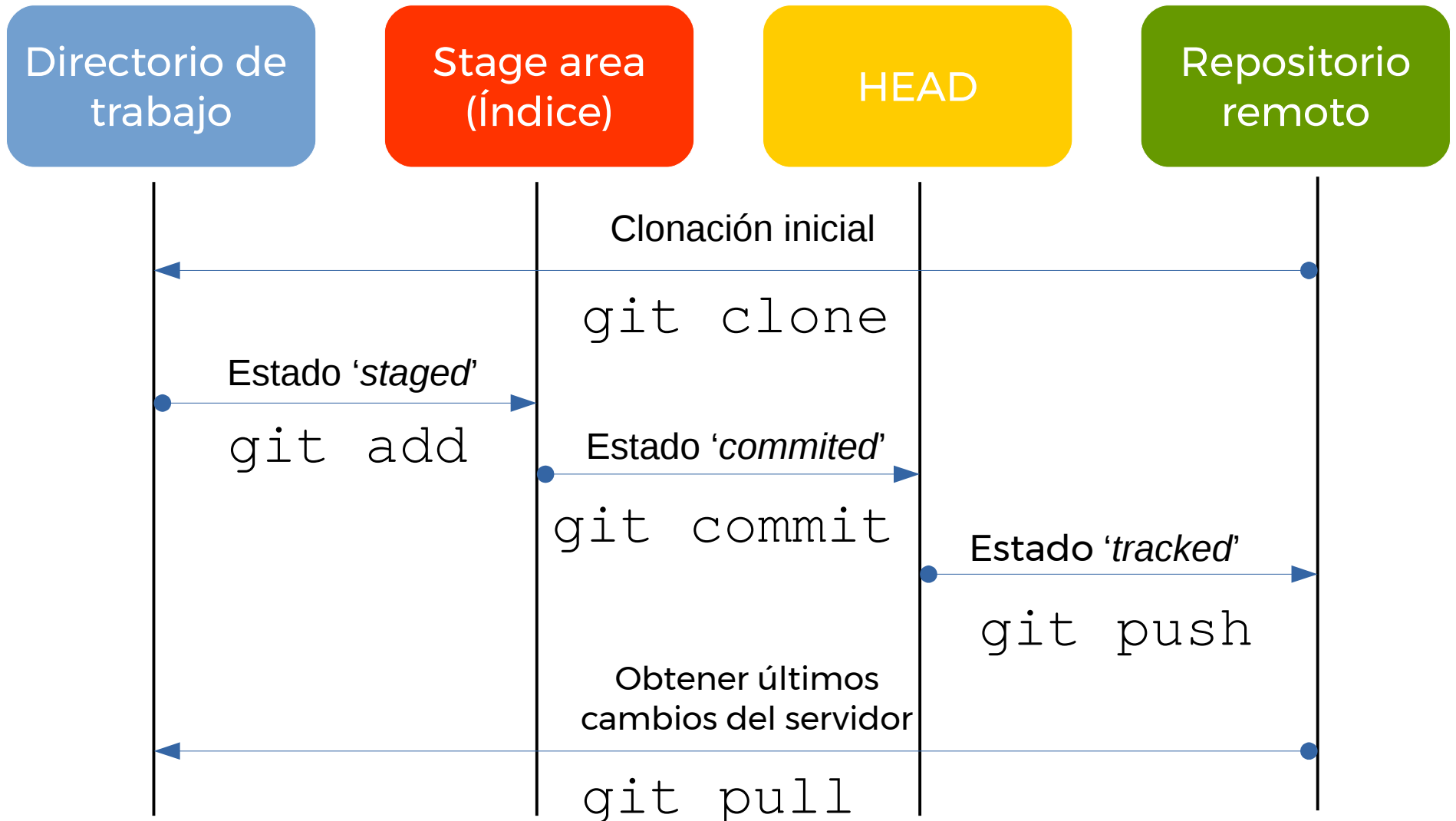
```
# git config -global user.name  
"Oscar Delgado"
```

```
# git config -global user.email "  
oscar.delgado@uam.es"
```

- Almacenar credenciales localmente:

```
# git config credential.helper store
```

# Flujo de trabajo



# Añadiendo cambios

- Una vez modificado un fichero, se añade al próximo *commit* con:

```
# git add <nombre_fichero>
```

```
# git add *
```

- Comprobad los cambios con:

```
# git status
```

# Preparando el envío

- Todos los ficheros agrupados en el stage area se envían al servidor (“hacer un *commit*”) con:

```
# git commit -m "Mensaje"
```

- Ahora los cambios están en HEAD, pero todavía NO en el repositorio remoto.

# Enviando los cambios al servidor

- Finalmente, para enviar el último commit al servidor:

```
# git push origin master
```

- Si no hubiéramos clonado el proyecto de Gitlab, tendríamos que conectar nuestro repo local a un servidor remoto con:

```
# git remote add origin <servidor>
```

# Recibiendo los últimos cambios

- Para actualizar el repo local con los últimos cambios del servidor, ejecutar:

```
# git pull
```

- Si surgen conflictos, resolver manualmente, y marcarlos como resueltos con:

```
# git add <fichero>
```

# Comandos útiles

- **Borrar ficheros:**

```
# git rm <fichero>
```

- **Ver historial de cambios:**

```
# git log
```