



# Compiladores 2017-1

Dra. Elisa  
Pedro  
Diana



# Presentación del Laboratorio

- Forma de trabajo
  - En equipo
  - Código en repositorio git
- Grupo

<https://goo.gl/C0GGuK>

# Forma de trabajo

- Prácticas y proyectos serán realizados en equipo de 3 personas.
- El código será revisado y calificado desde el repositorio.

# Grupo de Google

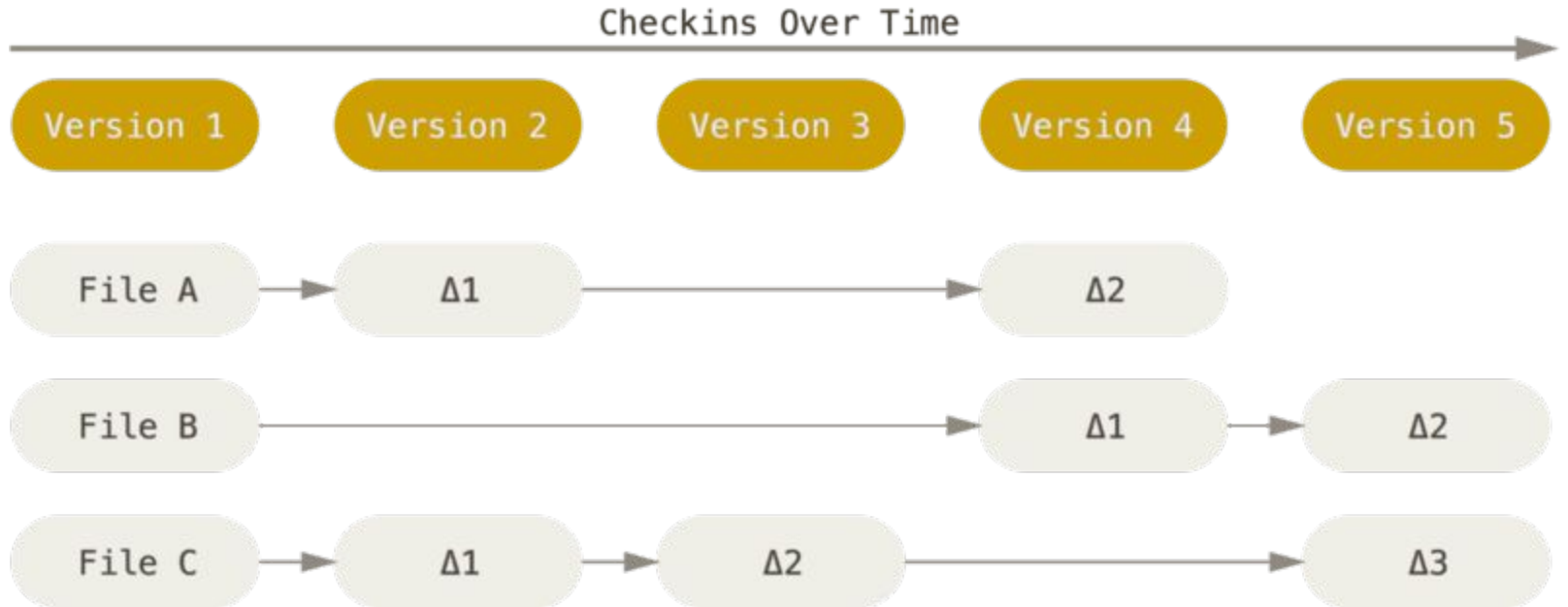
- Avisos, comentarios y dudas.
- Archivos de prueba
- Dirección: <https://goo.gl/C0GGuK>

# Git

- Manejador de versiones.
- Si ustedes tuvieran que implementar un manejador de versiones, ¿cuál sería su estrategia a seguir?

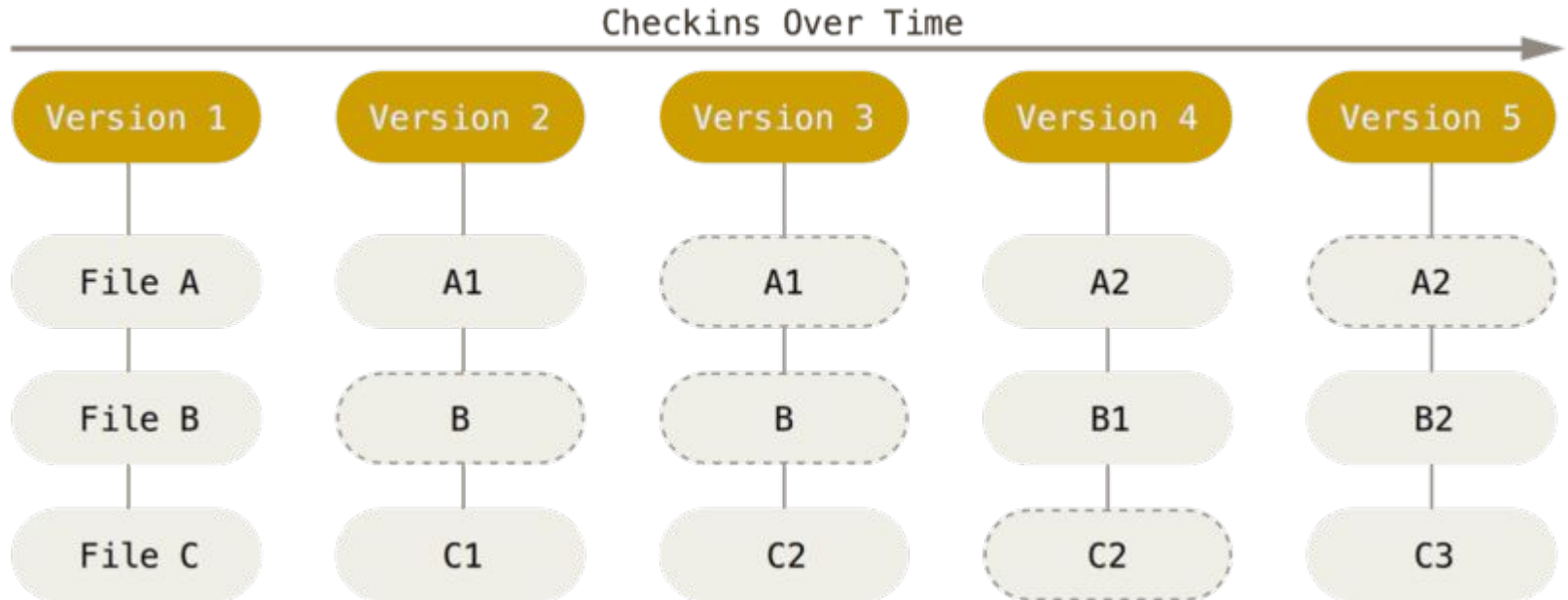
# Estrategias en el mercado

- Archivos y serie de diferencias



# Estrategias en el mercado

- Snapshots



# Creador





# Ventajas de Git

- Puede trabajar todo de manera local
- Tienes la historia completa del proyecto en tu base de datos local
- Puedes calcular las diferencias entre los archivos que se editaron hace un mes y los actuales.
- No necesitas conexión latente a internet.
- Provee integridad a través de checksumming (Sha-1):
  - Cada Snapshot tiene un identificador de 40 dígitos.
- Aún siendo lo más descuidado posible... Es difícil borrar datos.
- Te da tu propio espacio, que puedes convertir en un chochinero sin ninguna consecuencia.

# Etapas de almacenamiento

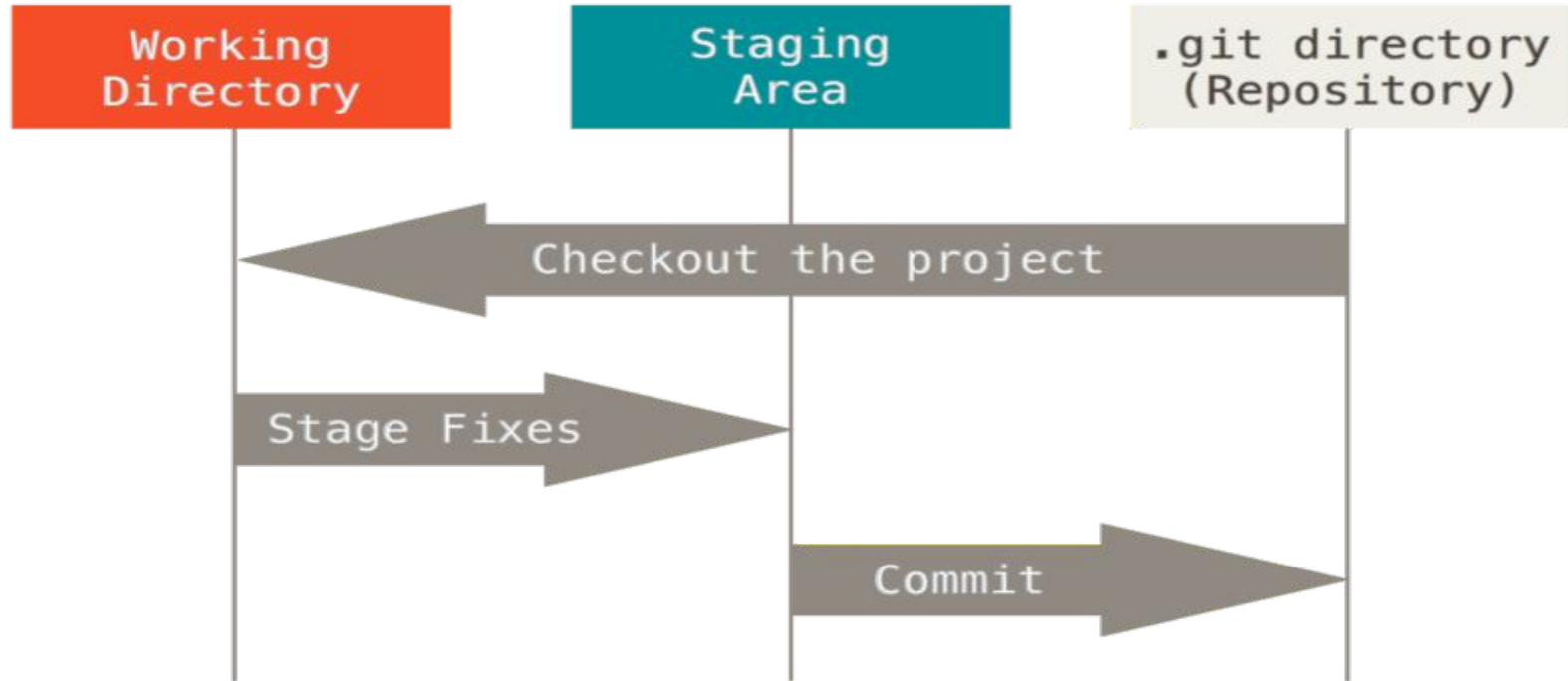
Hay 3 partes dónde se pueden guardar cambios:

1. Directorio de trabajo - Versión del repositorio guardada localmente en disco y disponible para su modificación.
2. Área de presentación.
3. .git - Directorio de git .

Dependiendo dónde si y dónde no haya sido guardado, recibe los siguientes nombres:

1. Modificado (Modified)
2. Presentado (staged)
3. Comprometido (Committed)

# Etapas de almacenamiento



# Comandos

- clonar : **git clone** [repositorio]
- agregar: **git add** [ . | \* | ruta\_archivo]
- ver historial de cambios: **git log**
- actualizar repositorio: **git pull**
- commit: **git commit -m "Mensaje"**
- empujar cambios: **git push**

# Relación con el repositorio

- Conflictos: cuando mis cambios sucedieron en los mismos archivos y líneas de la versión principal
- Solucionando conflictos.

# Recomendaciones

- Tratar de entender el funcionamiento.
- No dejar pasar mucho tiempo sin jalar cambios.
- No dejes mucho tiempo sin empujar cambios
- Leer
- .gitignore

# Práctica 0

- Entrar a GitLab <https://gitlab.com/>
- Crear una cuenta, si es que aún no la tienen.
- Formar equipos de 3.
- Crear un repositorio para el curso
- Agregar las cuentas de los 3 integrantes y la cuenta dianamontes1516
- Crear un README.md que tenga en nombre del equipo.
- Cada uno de los integrantes editará desde su equipo el archivo, poniendo su nombre y correo.

# Práctica 0

- El repositorio deberá tener dos carpetas, una que se llame Prácticas y otra que se llame Proyectos, en la carpeta de Prácticas crearán una carpeta llamada Practica0.
- Crear un archivo llamado practica0.[txt | org | md]. En el contestarán las siguientes preguntas:
  - Explica cómo se crea el identificador de 40 dígitos para el commit.
  - Escribe el comando para regresar al commit 911e8c9
  - Describe lo que son las ramas, para qué sirven y los comandos relacionados con ellas. (git branch, git merge, ) explicando su funcionalidad.