



GeoGig

Luis Octavio Ramírez Fernández, Instituto de Geografía UNAM

1 de diciembre de 2014

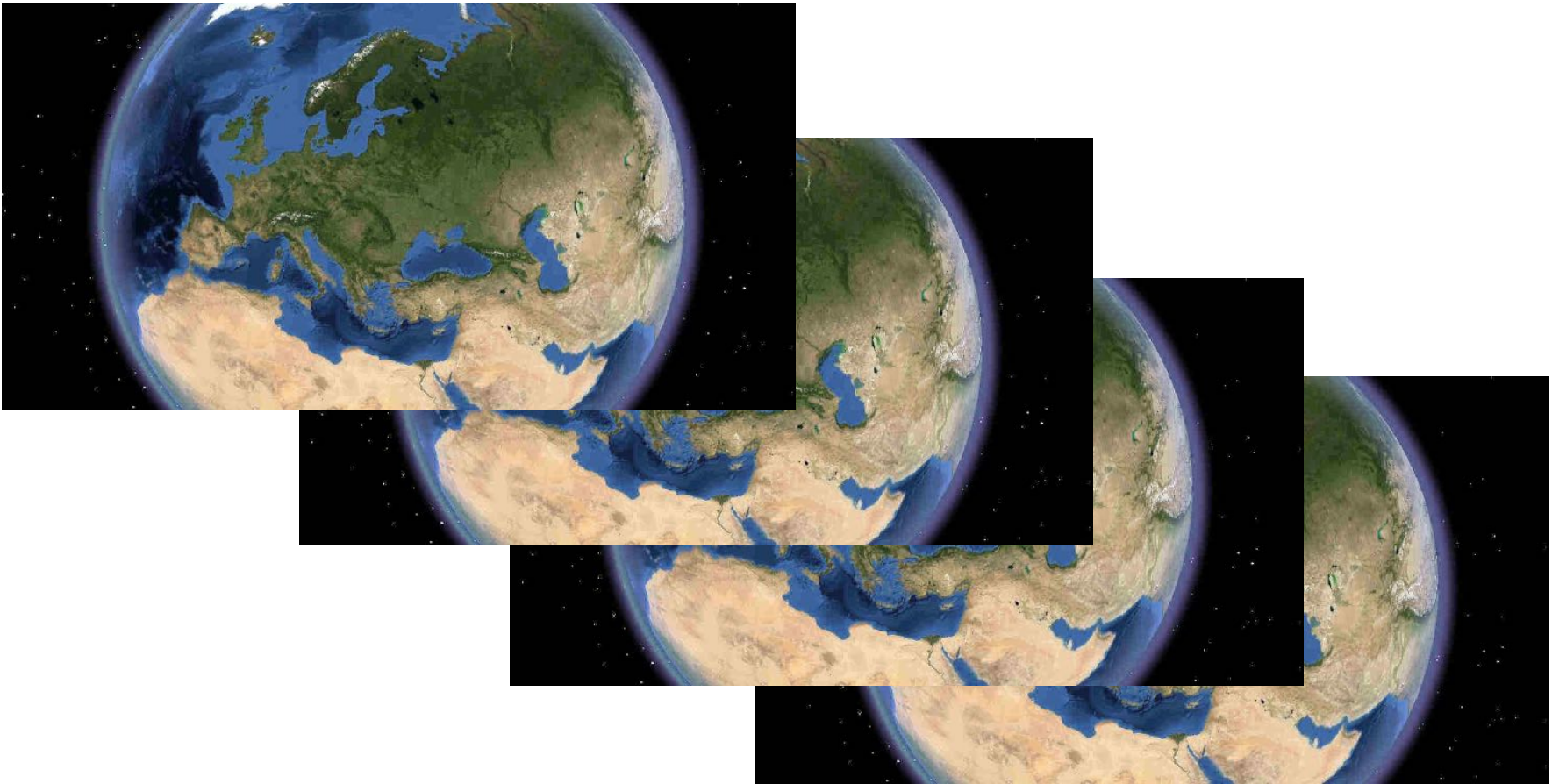


Datos Espaciales





... edición de histórica





git



GeoGig != git





```
git() {  
  administra(archivos)  
  organiza(directorios)  
  almacena(snapshots)  
}
```





```
-git() {  
+geogig {  
-  administra(archivos)  
+  administra(rasgos)  
-  organiza(directorios)  
+  organiza(capas)  
  almacena(snapshots)  
-}
```





¿Qué es GeoGig?

Es un sistema de control de versiones distribuido diseñado especialmente para datos espaciales.





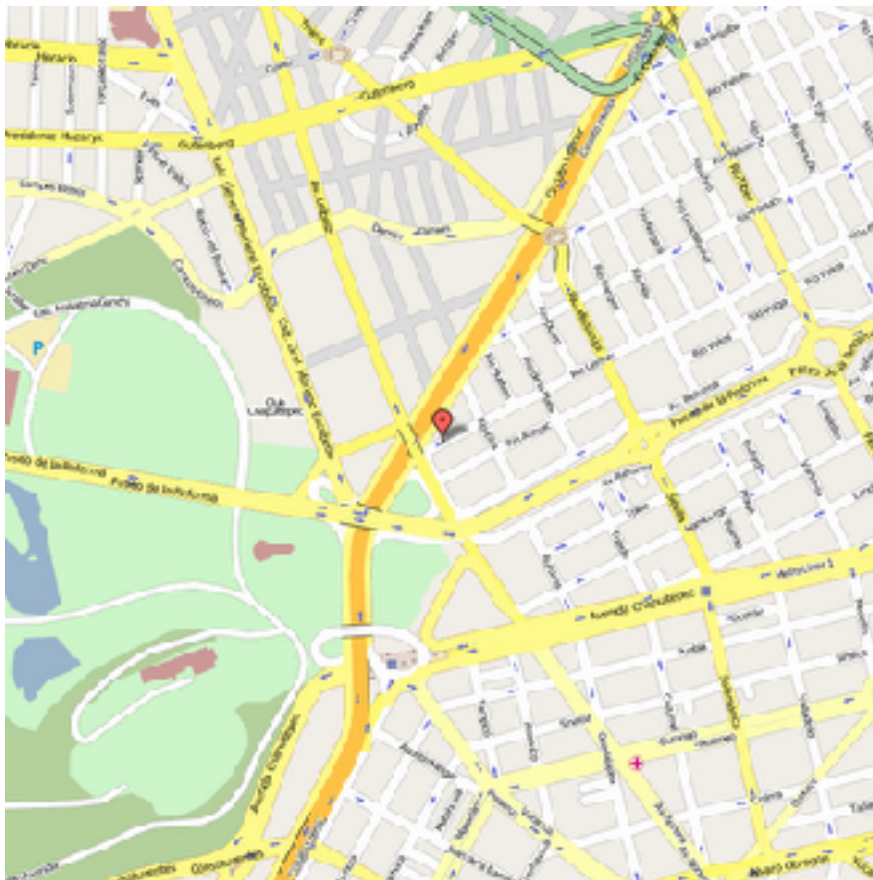
¿Qué es GeoGig?

- Repositorio de datos espaciales distribuido y colaborativo.
 - “geogig no es git”
- Características de GIT:
 - Distribuido y trabajo local.
 - Versioning, comparativo y uniones.
 - Push/pull.
- Licencia BSD





Características de GeoGig



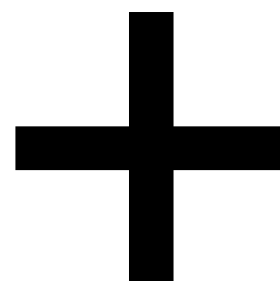
- Almacena datos espaciales
- Rápido
- Contenedor de versiones compacto.
- Revisión de datos históricos.
- Resuelve conflictos.



Formatos soportados GeoGit

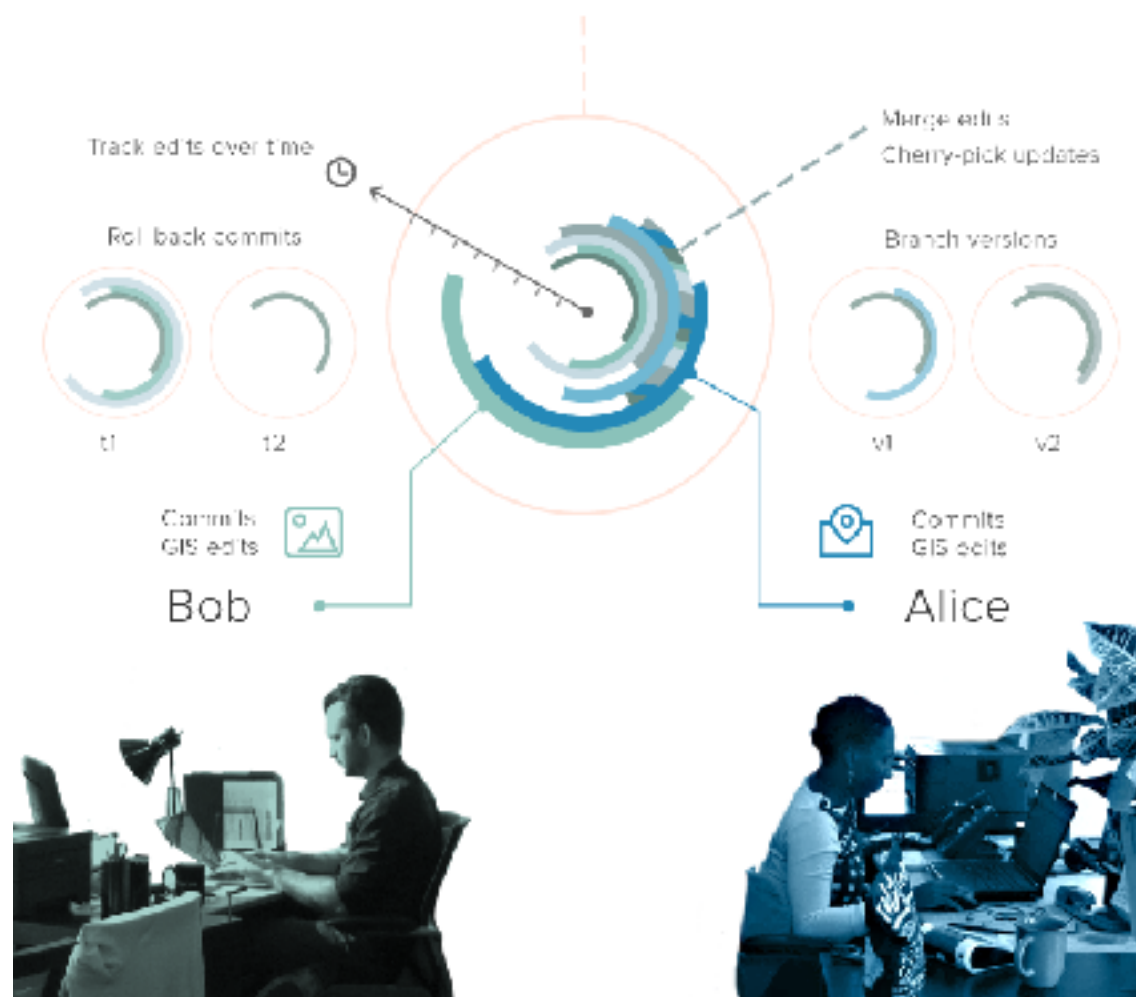


- Shapefiles
- SpatialLite / SQLite
- PostGIS / PostgreSQL
- OSM / XML





A Shared GeoGig Repository

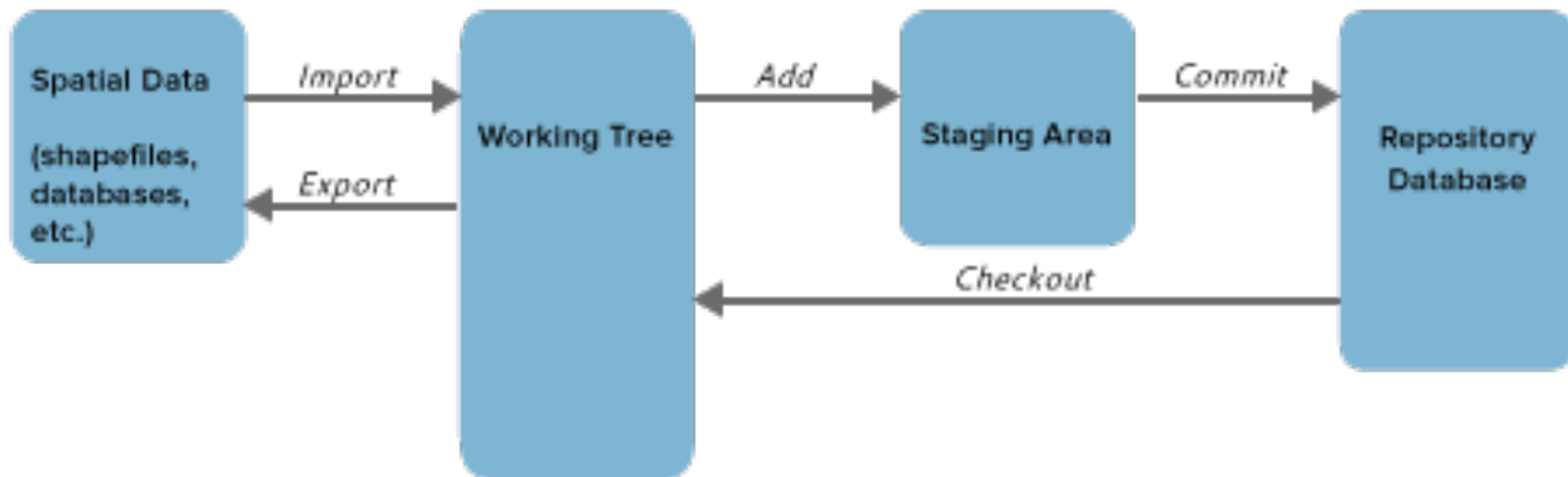


GEOGIG WORKFLOW



Flujo de trabajo local

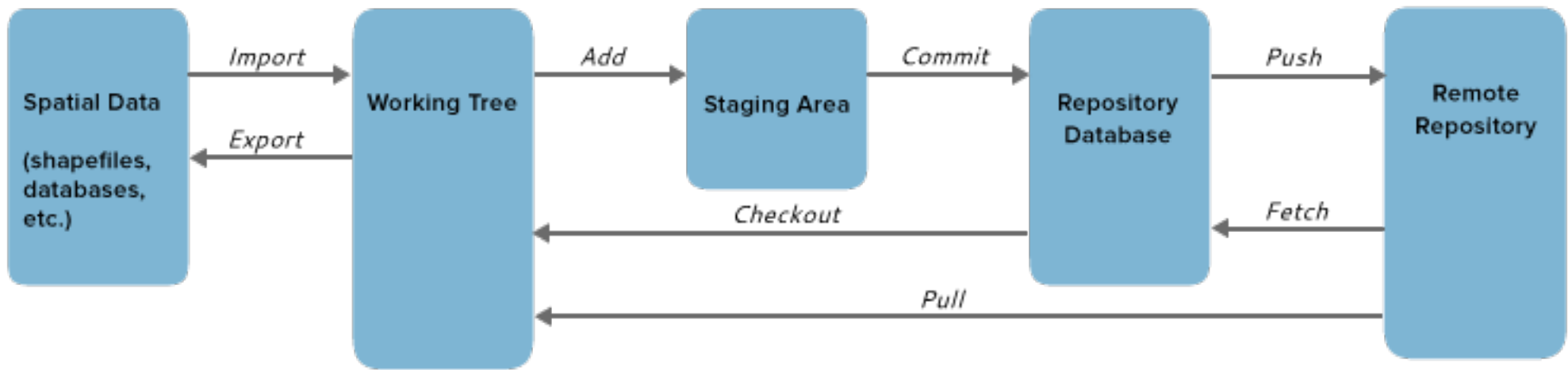
GeoGit Workflow





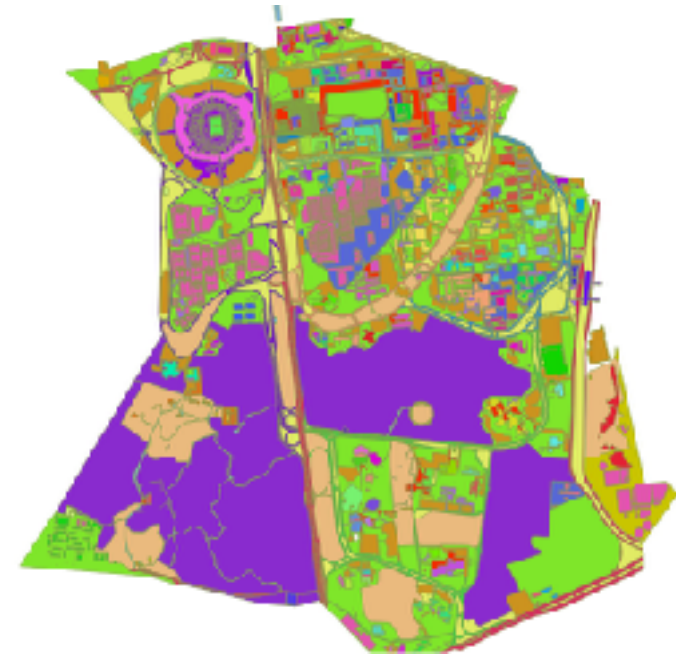
Flujo de trabajo colaborativo

GeoGit Workflow (+ Remote Repository)





Flujo de trabajo local





Flujo de trabajo con GeoGig



Configuración básica

```
geogig config --global user.name "Juan Pérez"  
geogig config --global user.email juan@example.com
```



Iniciando con geogig

Crear un repositorio local

```
$ mkdir geogig_trabajo
```

```
$ cd geogig_trabajo
```

```
$ geogig init
```



Importando y agregar datos

Importar

```
$ geogig shp import ruta/archivo.shp
```

Agregar datos

```
$ geogig add
```

Listar contenido

```
$ geogig ls
```




Trabajando con geogig

Estado del repositorio

\$ geogig status

Confirmar cambios

\$ geogig commit -m “Primer cambio”

Revisar confirmaciones

\$ geogig log



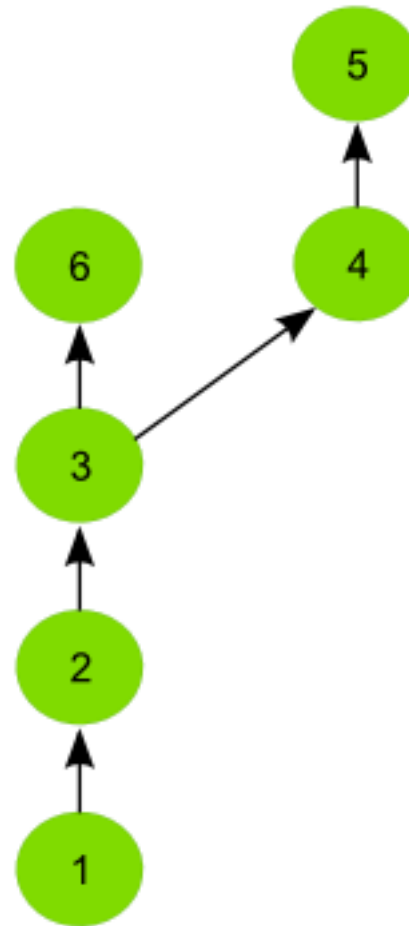
Trabajando con geogig

Crear la rama en el punto actual. Es necesario hacer checkout a la misma.

```
$ geogig branch <nombre>
```

Crea la rama a partir del commit dado. Es necesario hacer checkout.

```
$ geogig branch <nombre> <COMMIT>
```





Trabajando con git

No toca los cambios locales

\$ geogig checkout <COMMIT>

Cambia entre ramos

\$ geogig checkout <rama>



Trabajando con geogig

Fusiona la rama indicada en la rama actual

```
$ geogig merge <nombre_rama>
```

Muestra los ficheros conflictivos y las diferencias

```
$ geogig diff HEAD~1 HEAD
```



Trabajando con geogit

Abortar el proceso y volver a la situación anterior al intento de merge

```
$ geogit reset --hard HEAD
```

Deshacer si ya se había confirmado con git commit

```
$ geogit reset --hard ORIG_HEAD
```




Trabajando con geogig

\$ geogig tag -s <nombre> -m <mensaje>

Mostrar lista de tags

\$ geogig tag

Mostrar lista y descripción

\$ geogig tag -n



Trabajando con git

Clonar local

```
$ geogig clone geogig_local clone_geogig
```



Referencias

1. Geogit: Versioned storage for geospatial data (2013). FOSS4G 2013, Nottingham. Recuperado el 30 de noviembre de 2014 de <http://boundlessgeo.com/wp-content/uploads/2013/05/foss4g2013-sponsor-day-geogit.pdf>
2. Winslow, D (2013). Inside GeoGit. FOSS4G 2013, Nottingham. Recuperado el 30 de noviembre de 2014 de <http://boundlessgeo.com/wp-content/uploads/2013/10/geogit-foss4g2013.pdf>
Geogit 1.0 Workshop. Recuperado el 30 de noviembre de 2014 de <http://geogig.org/workshop/index.html>
3. GeoGig 1.0 User Manual, recuperado el 30 de noviembre de 2014 de <http://geogig.org/docs/index.html>
4. boundlessgeo (2012). GeoGig. Github. Recuperado el 30 de noviembre de 2014 de <https://github.com/boundlessgeo/GeoGig>



¿Preguntas?



¡ Gracias !



Contacto:

Luis Octavio Ramírez Fernández.
lr Ramirez@igg.unam.mx