



Test de Manejo

ROS y más ROS

11 de Abril de 2018

Hora de las noticias

Nicolás Consuegra(Ing Trainee)

&

Valentina Castro(Ing Trainee)



STAN: THE SELF-DRIVING VALET

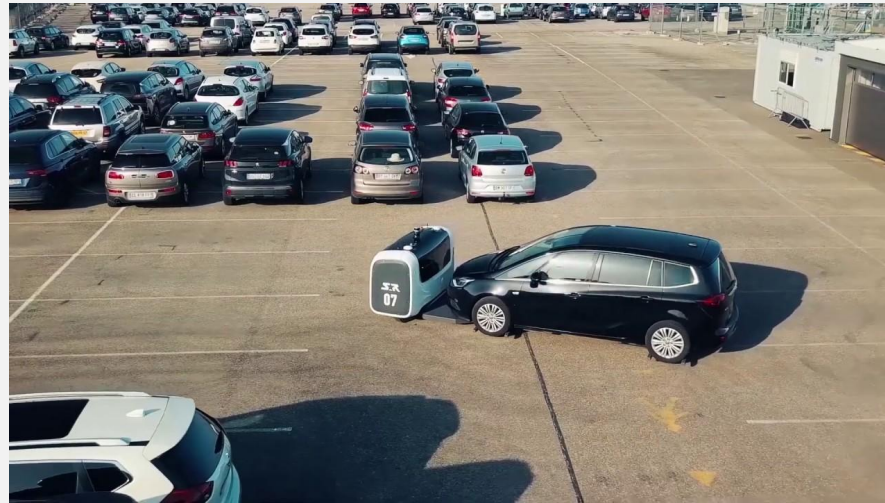
Nicolás Consuegra

Valentina Castro

Duckietown Chile

ASÍ TRABAJA STAN

- <https://youtu.be/ZYVXGYHw8Eg>



- “El robot usa la misma tecnología que los vehículos autónomos; es capaz de escanear su entorno en cualquier momento, y reaccionar a cualquier cosa que pudiera estar en el camino.”
(Stéphane Evanno, COO de Stanley Robotics, 2017)



BENEFICIOS



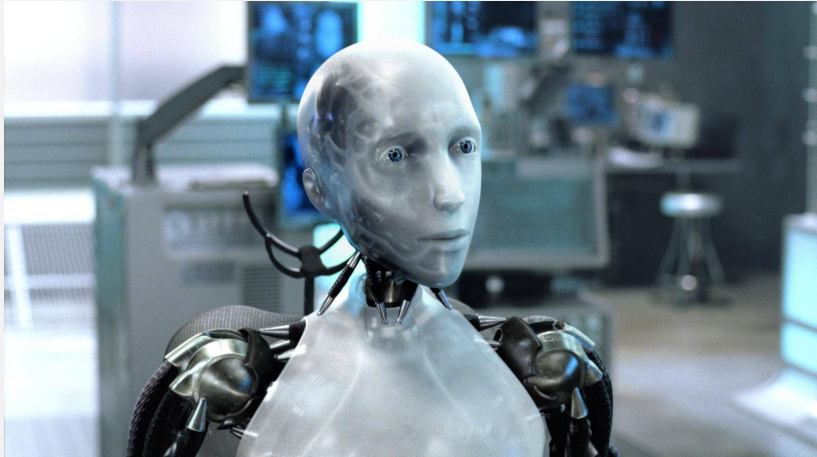
- Crea un 50% más de espacios en la misma área de superficie
- Mayor eficiencia en los tiempos
- Ahorro para los usuarios

¿EN QUÉ AFECTA A DUCKIETOWN?

- Mayor eficacia en el estacionado de los duckiebots
- Mejora en la disponibilidad de transporte



REFLEXIÓN



- Adaptación de los robots en la sociedad
- Enfoque de la ciencia en cuanto al uso de los avances tecnológicos





Test de Manejo

ROS y más ROS

12 de Abril de 2018

Capacitación de hoy

- Test de manejo
- Respaldo de cambios con Git
- Revisión del roadmap de próximas capacitaciones

Test de manejo

Test de manejo

Examen al final de la capacitación para obtener su licencia de conducir

Su duckiebot debe:

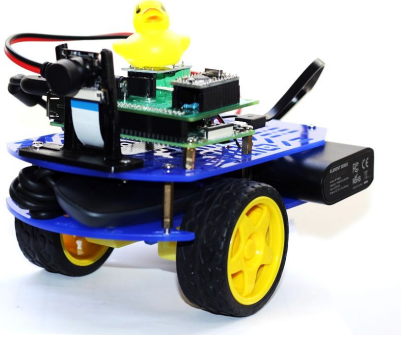
- Avanzar
- Retroceder
- Girar
- Tener un freno de emergencia



Respaldo de datos con Git

Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/
```



Documents



duckietown



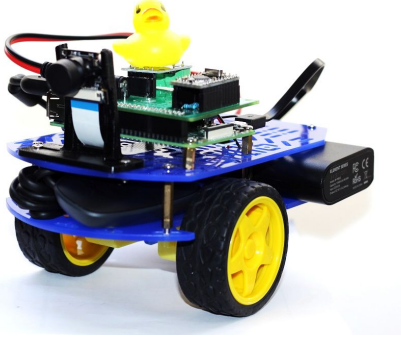
Pictures



Duckietown
Engineering
Chile

Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown
```



attic



catkin_ws

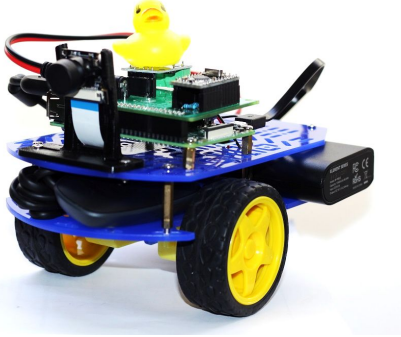


circuits



Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws
```



build



devel

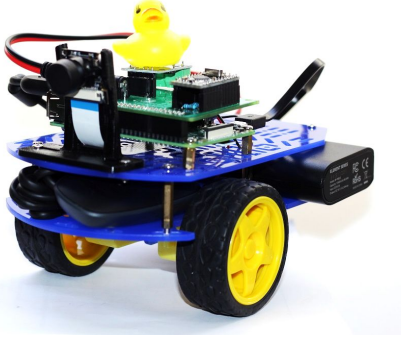


src



Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/src
```



ros_cap



ros_test

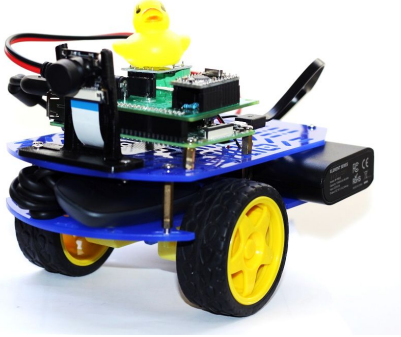


scene_segme
ntation



Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/ros_cap/src
```



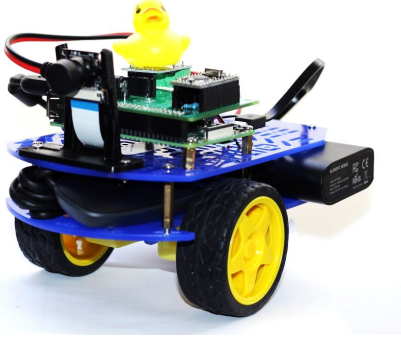
template.py



Duckietown
Engineering
Chile

Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/src/os_cap/src
```



template.py

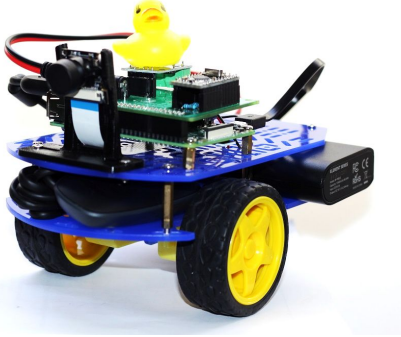


template1.py



Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/ros_cap/src
```



template.py



template1.py

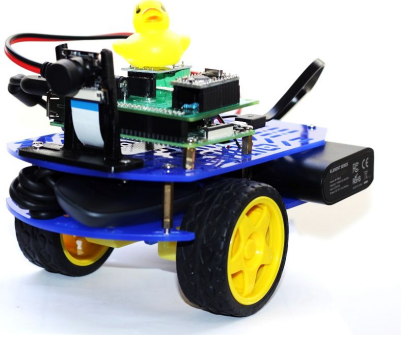


template2.py



Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/ros_cap/src
```



template.py



template1.py



template2.py



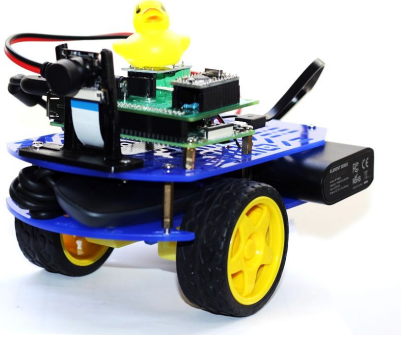
template_final.py



Duckietown
Engineering
Chile

Identificando cambios

```
duckiebot@duckiebot.local:/home/duckiebot/duckietown/catkin_ws/ros_cap/src
```



template.py



template1.py



template2.py



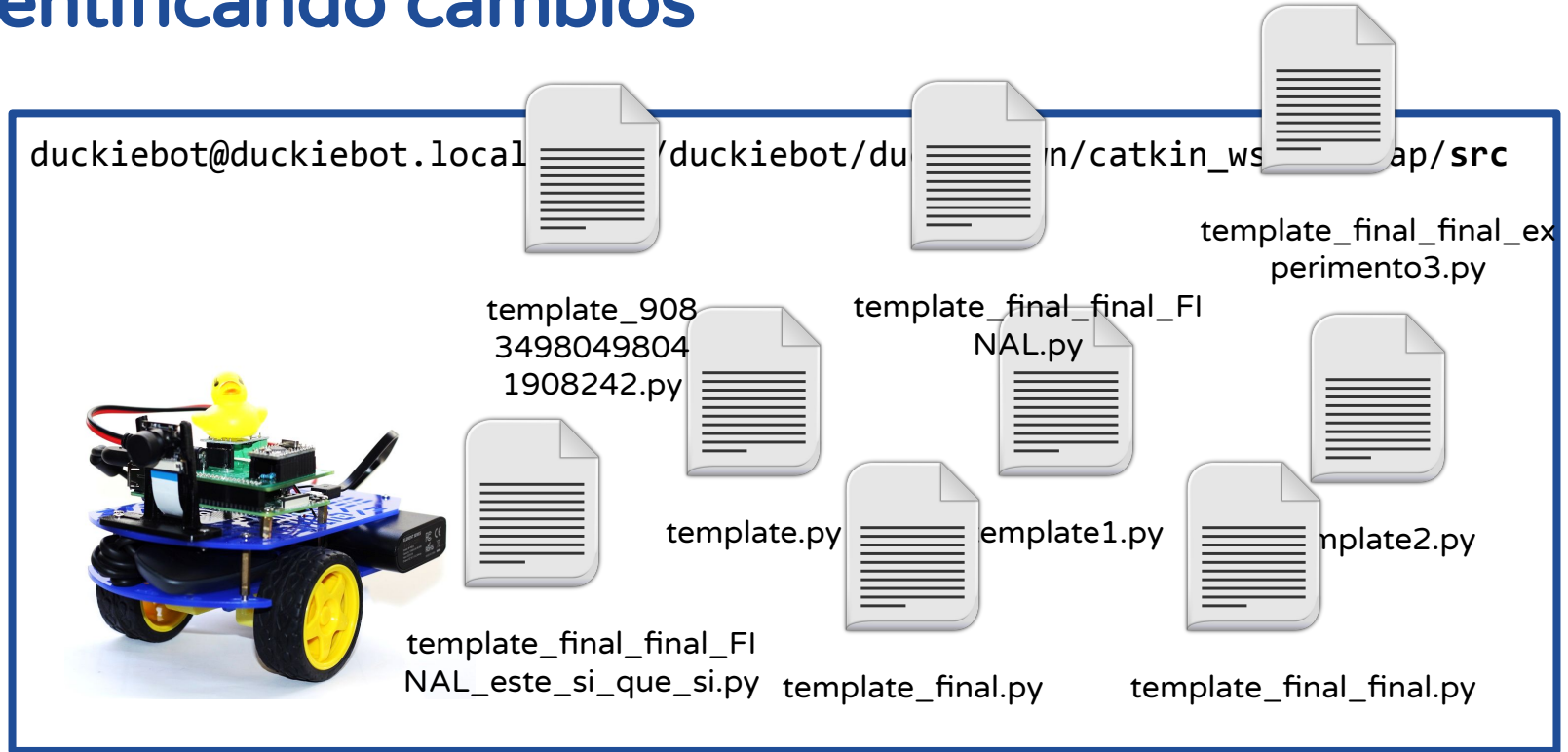
template_final.py



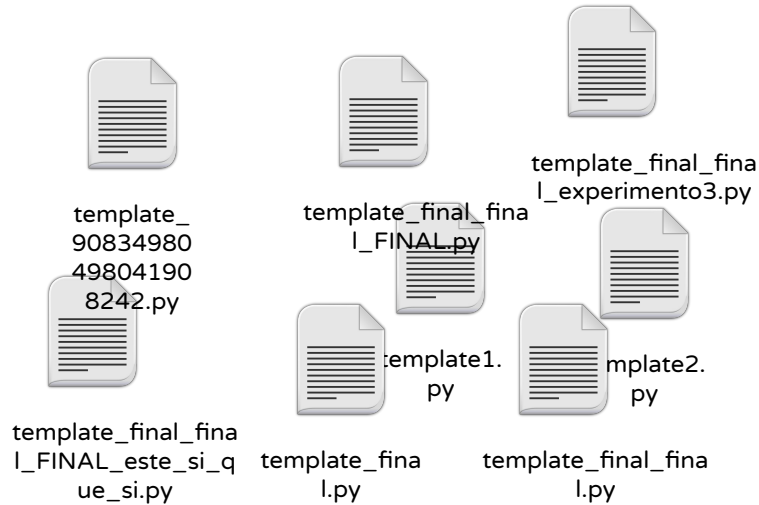
template_final_final.py



Identificando cambios



La ventaja de Git 1: Respaldos ordenados



Cómo respaldamos con Git

1. Conectarse al robot por cable
2. Abrir una terminal
3. Movernos a la carpeta que tiene Git configurado
 - a. La carpeta duckietown tiene una carpeta oculta `.git` que almacena toda la información (**¡no la borren!**)
4. Revisar si hay cambios en los archivos
 - a. `git status`
5. En rojo aparecen los archivos modificados o sin añadir al repositorio
6. Para añadir nuevos archivos, se usa
 - a. `git add archivo1.py archivo2.py archivo3.txt`



template.py



Cómo respaldamos con Git

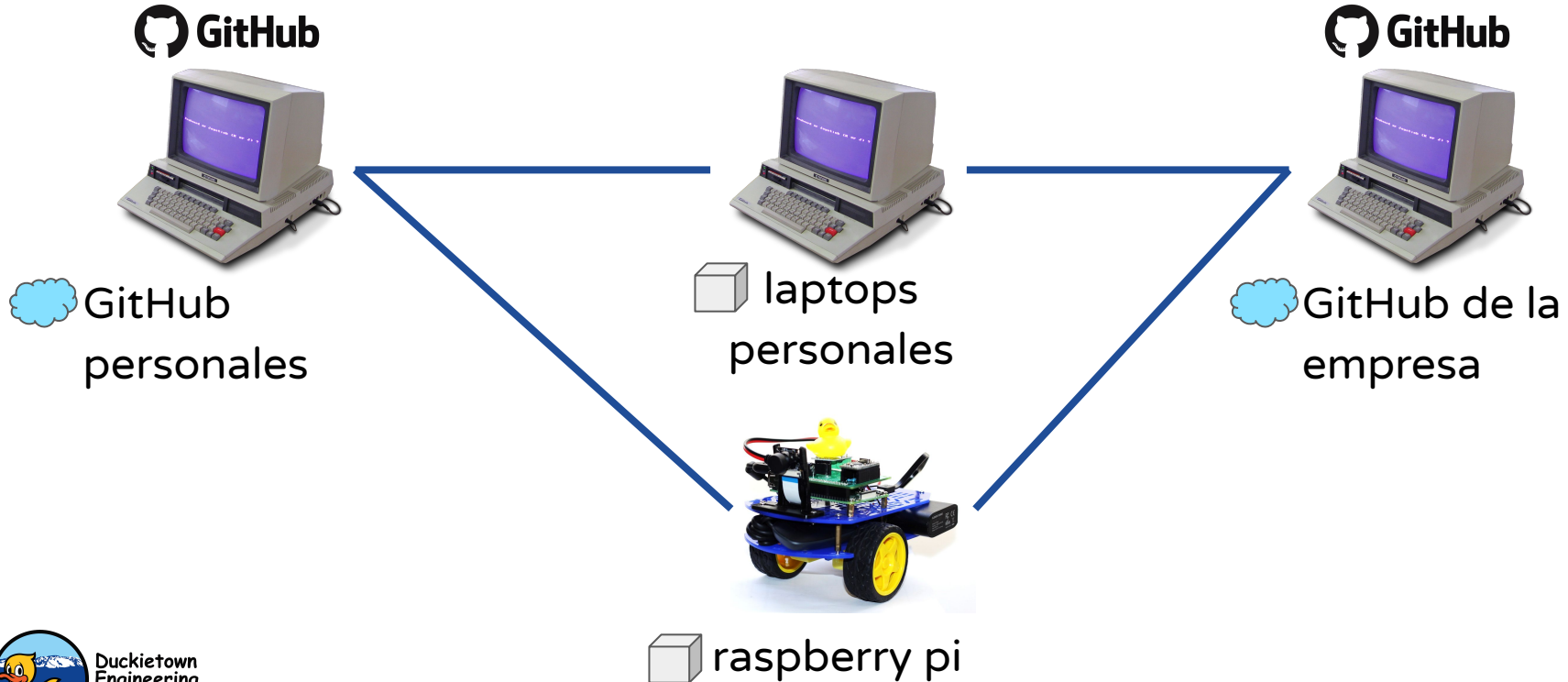
1. Si usamos `git status` nuevamente, los archivos añadidos aparecen en verde
2. Luego, debemos hacer un commit (respaldo), usando
 - a. `git commit -m "mensaje que informe sobre el cambio"`
 - b. Se recomienda que el mensaje sea claro y explicativo
 - c. No "asdsaasdsd", no "commit", no "jaja saludos"
3. Podemos revisar el historial usando
 - a. `git log`



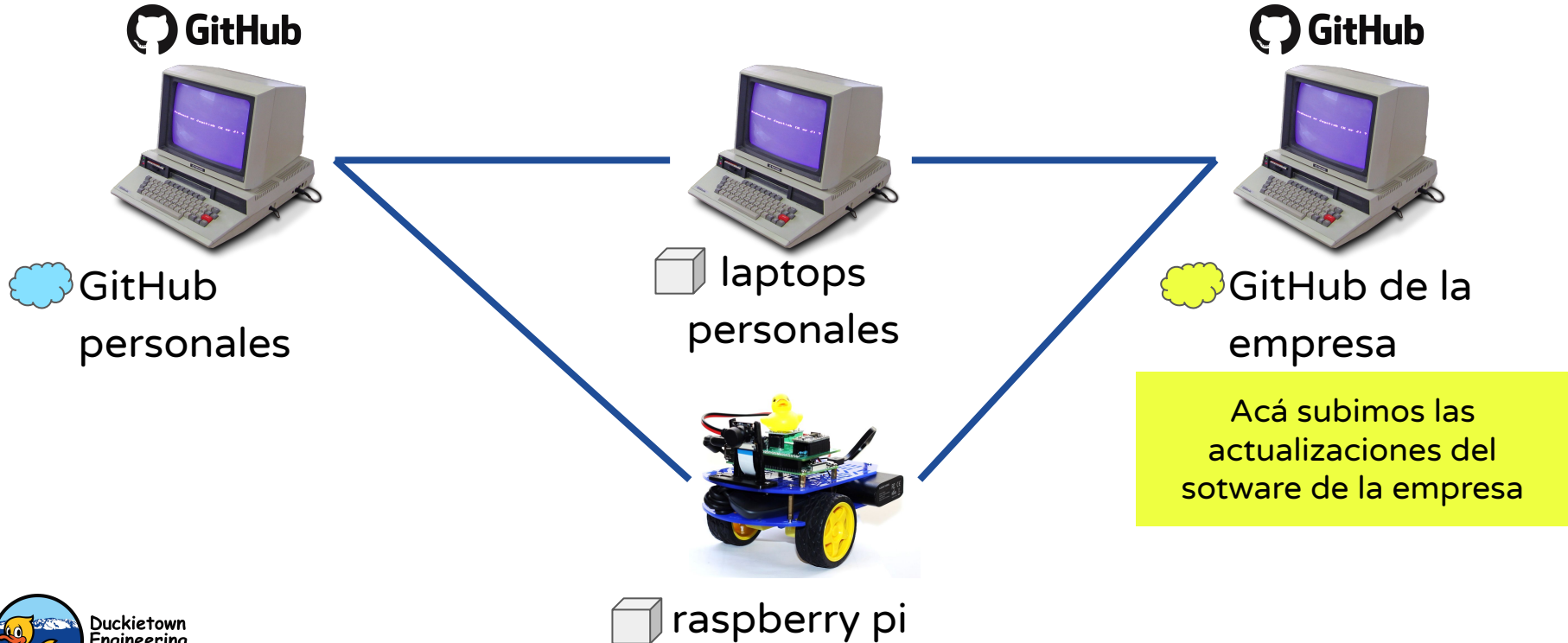
template.py



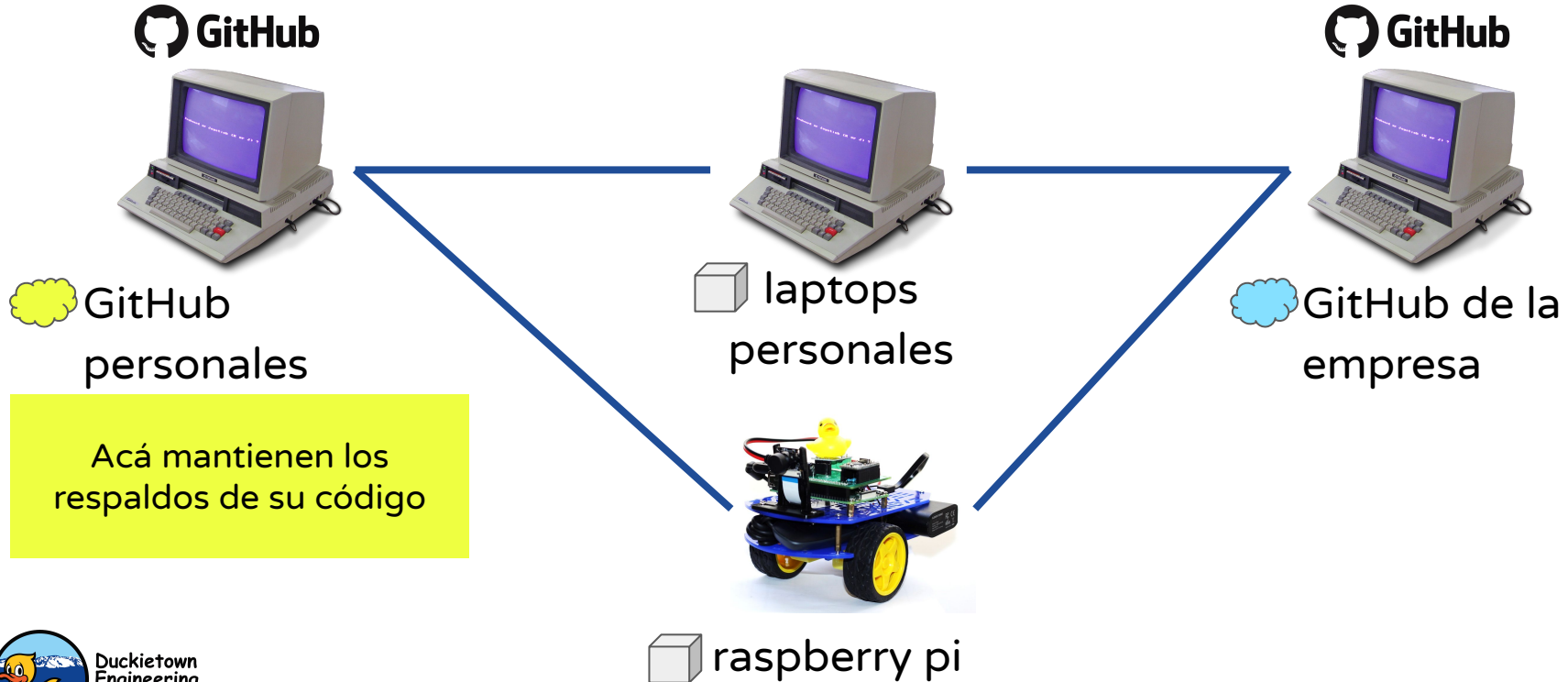
La ventaja de Git 2: Sincronización ordenada



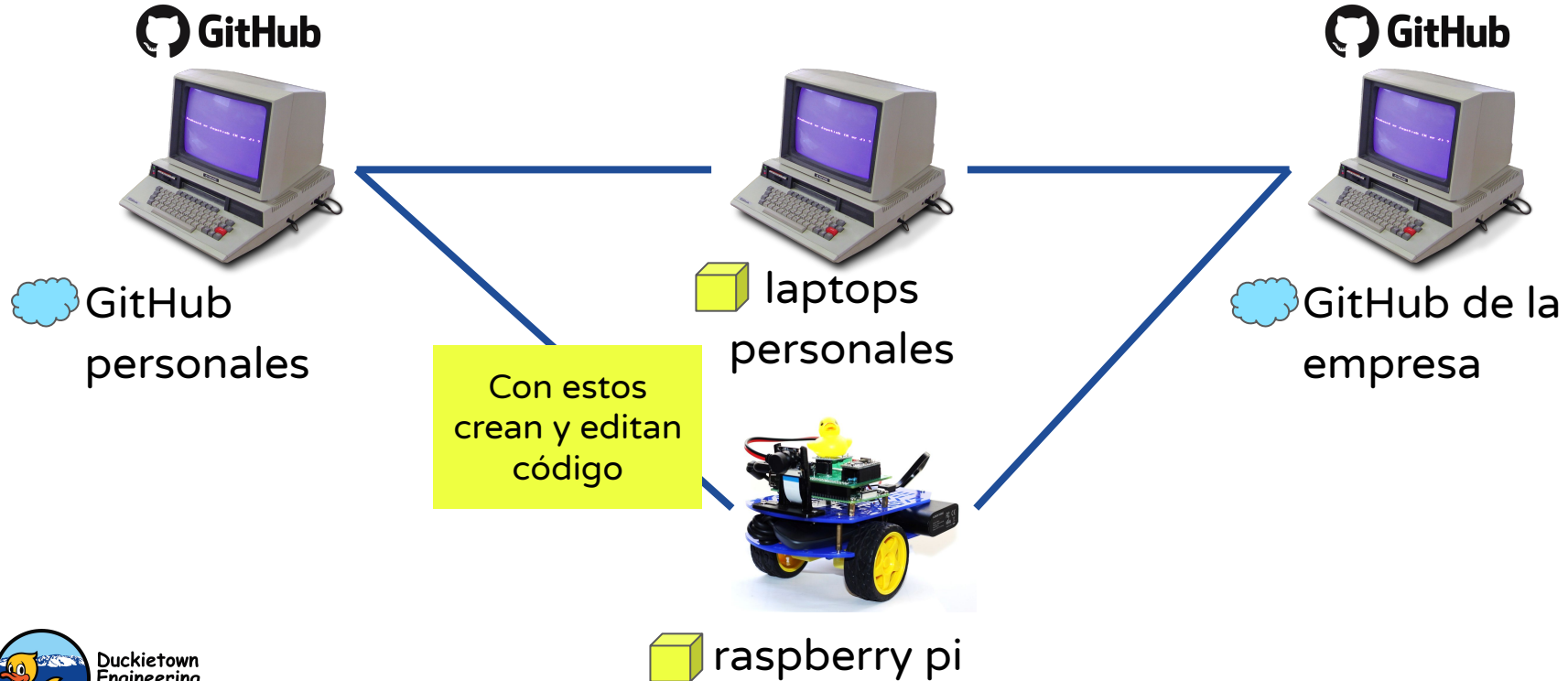
La ventaja de Git 2: Sincronización ordenada



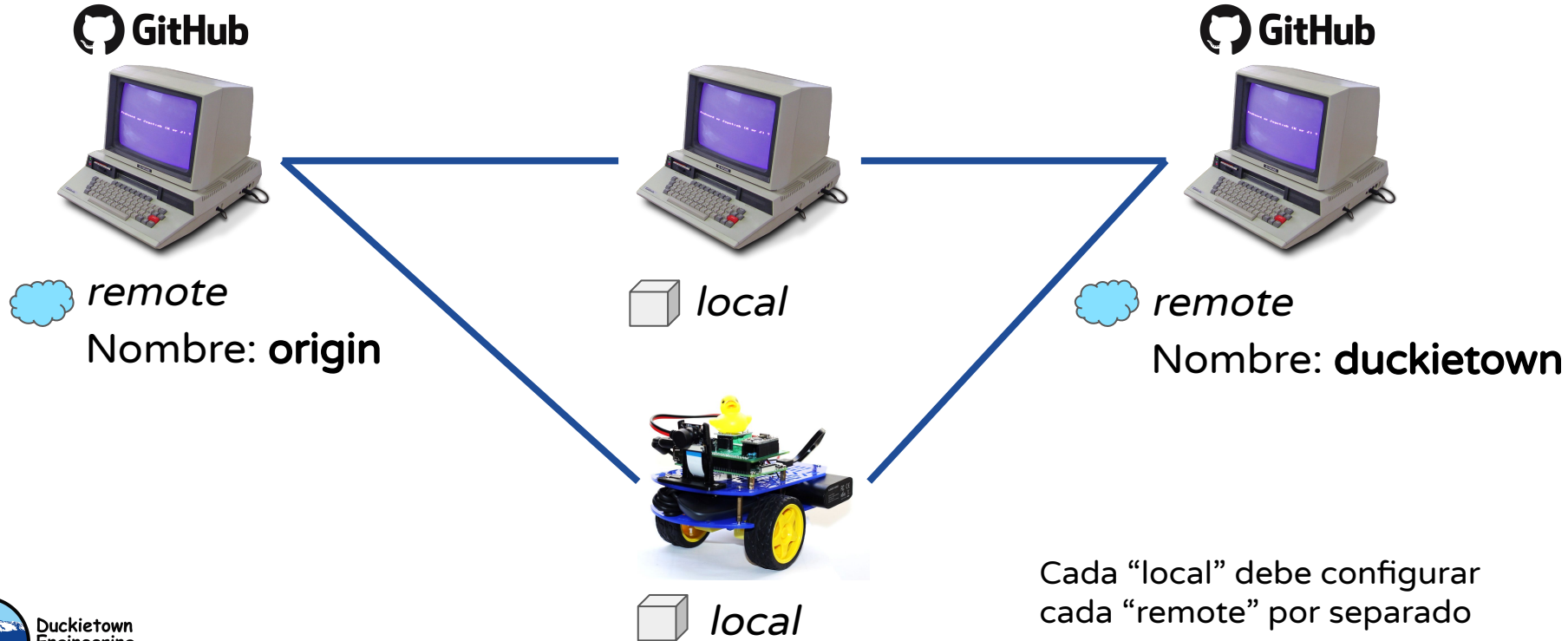
La ventaja de Git 2: Sincronización ordenada



La ventaja de Git 2: Sincronización ordenada



Nombres utilizados en cada computador



Cada "local" debe configurar cada "remote" por separado

Configuración inicial de Git

1. Configurar el nombre de usuario y email

- a. `git config --global user.name "Mona Lisa"`
`git config --global user.email "email@example.com"`

2. Revisar la configuración actual del repositorio

- a. `git remote -v`

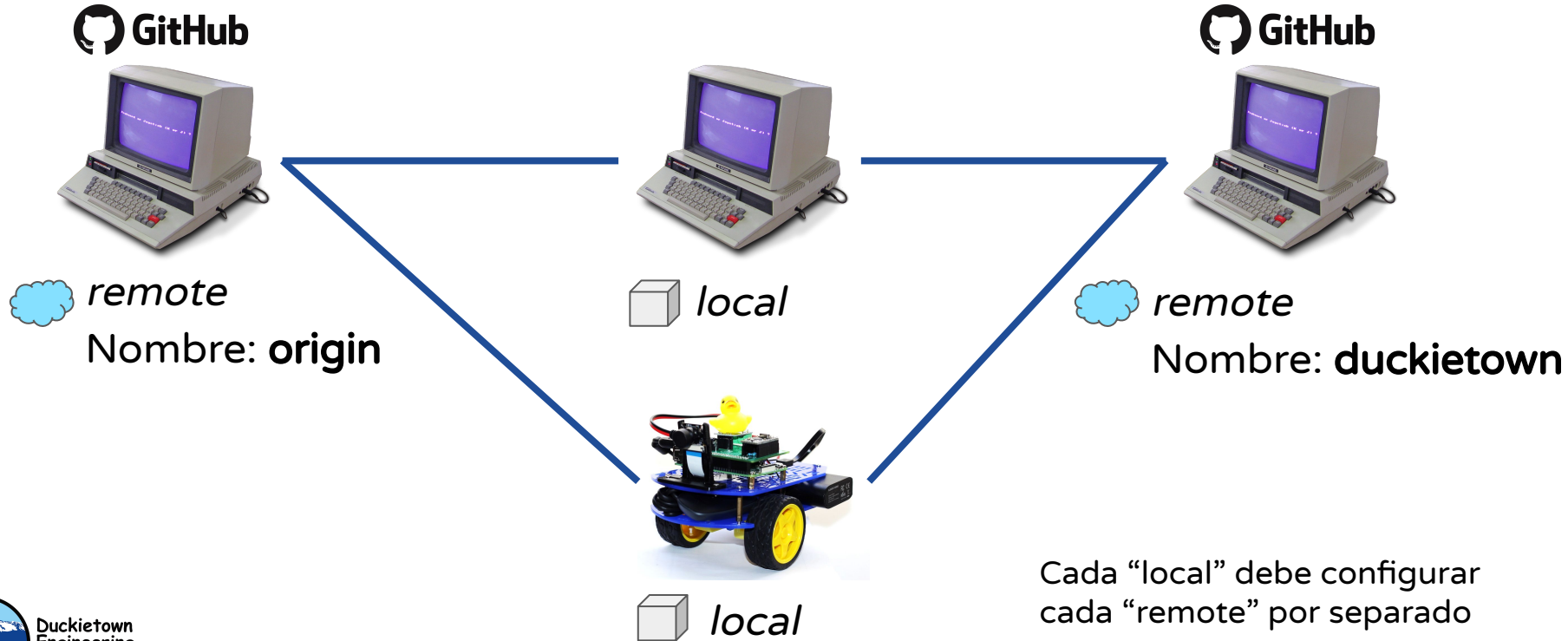
origin es el computador remoto con el que se sincroniza actualmente

Configuración de remotes

1. Configurar el repositorio del robot para que *origin* sea el fork personal y *duckietown* sea el de la empresa
 - a. `$ git remote set-url origin https://github.com/USERNAME/SU_FORK.git`
 - b. `$ git remote show origin`
2. Agregar el repositorio de Duckietown Chile como nuevo remote
 - a. `$ git remote add duckietown https://github.com/Duckietown-Chile/Software.git`
3. Confirmar que la configuración sea correcta
 - a. `git remote -v`



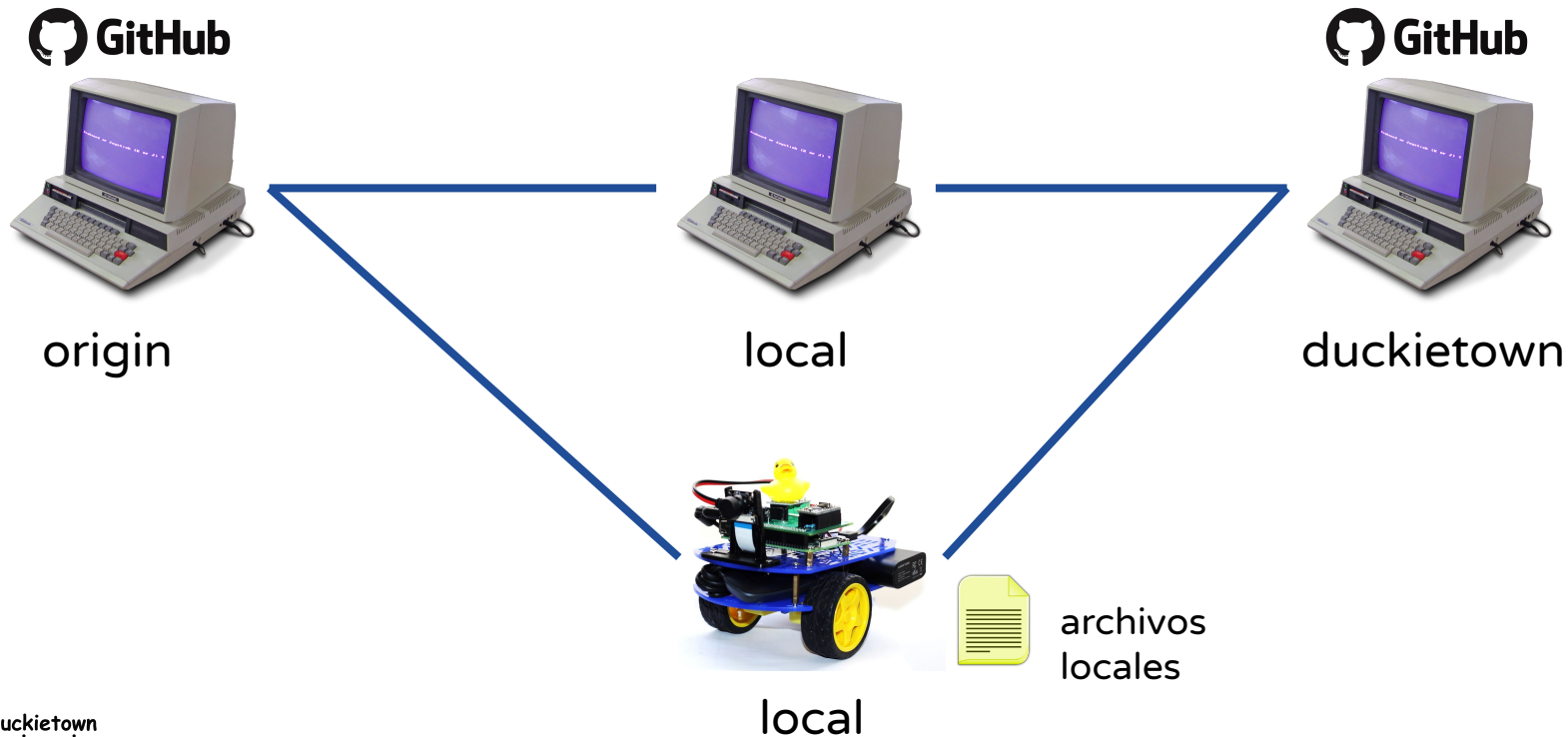
Nombres utilizados en cada computador



Cada “local” debe configurar cada “remote” por separado

Procedimiento de uso de Git en la empresa

Paso 1

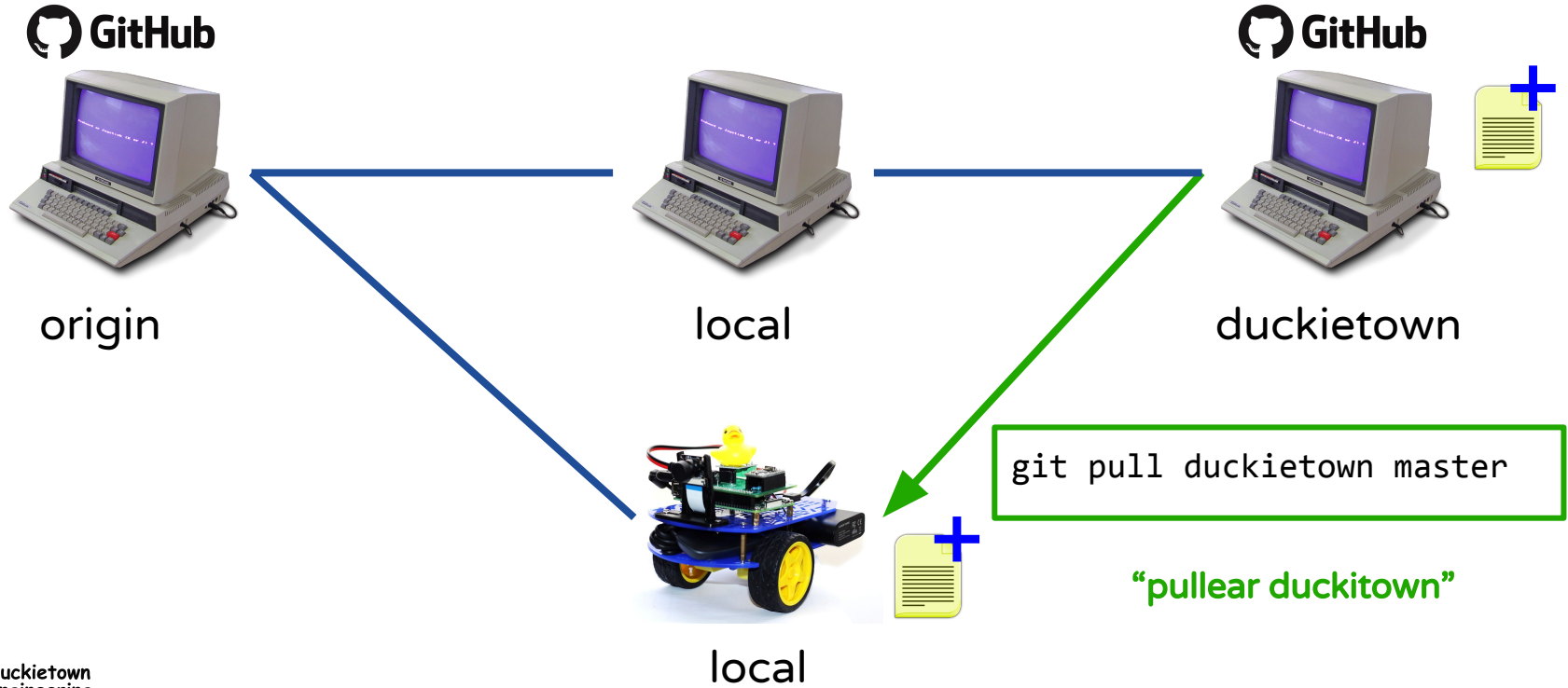


Paso 1

Antes de empezar a trabajar en el **robot**, revisar si hay actualizaciones de la empresa

a. `git pull duckietown master`

Paso 1



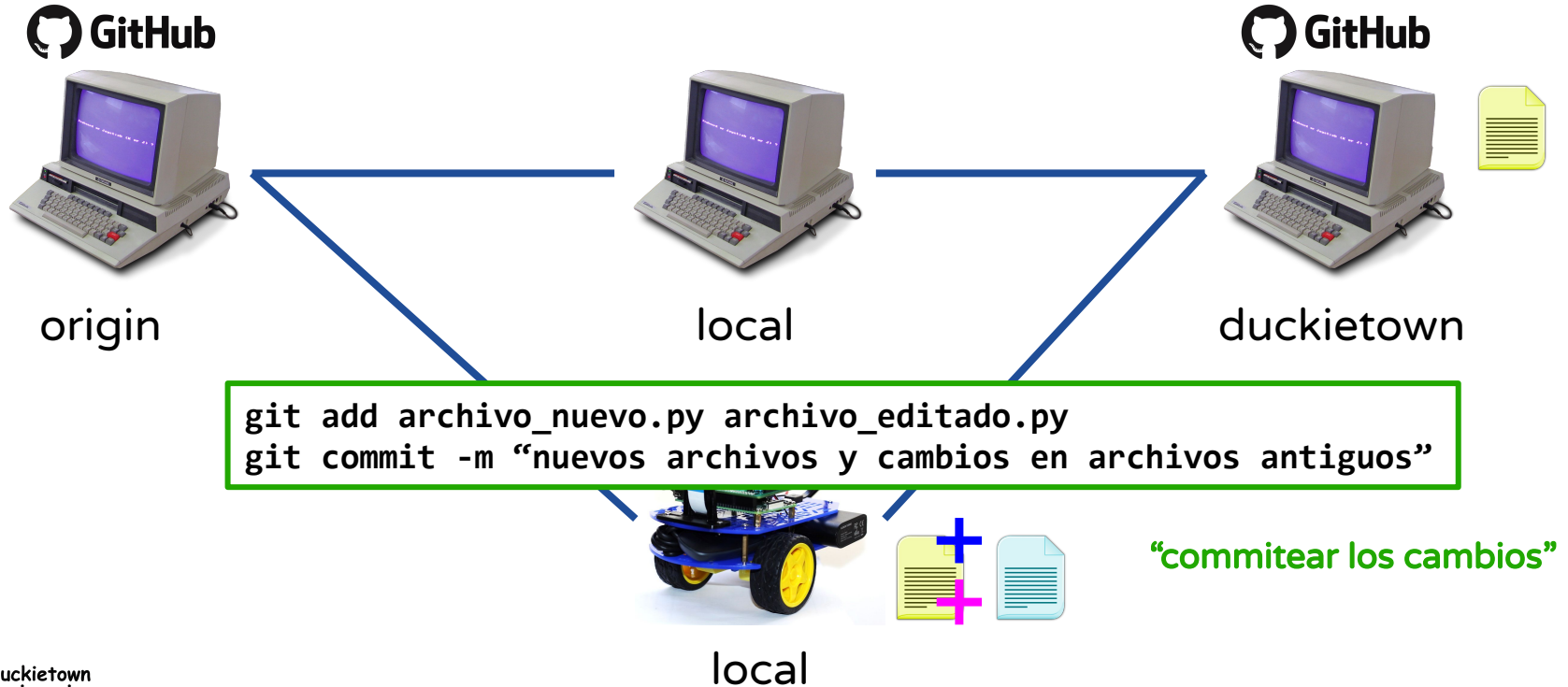
Paso 2

Editar, crear archivos nuevos, etc dependiendo de la tarea

Luego respaldar los nuevos cambios o nuevos archivos usando git

- a. `git add archivo_nuevo.py archivo_editado.py`
- b. `git commit -m "nuevos archivos y cambios en archivos antiguos"`

Paso 1

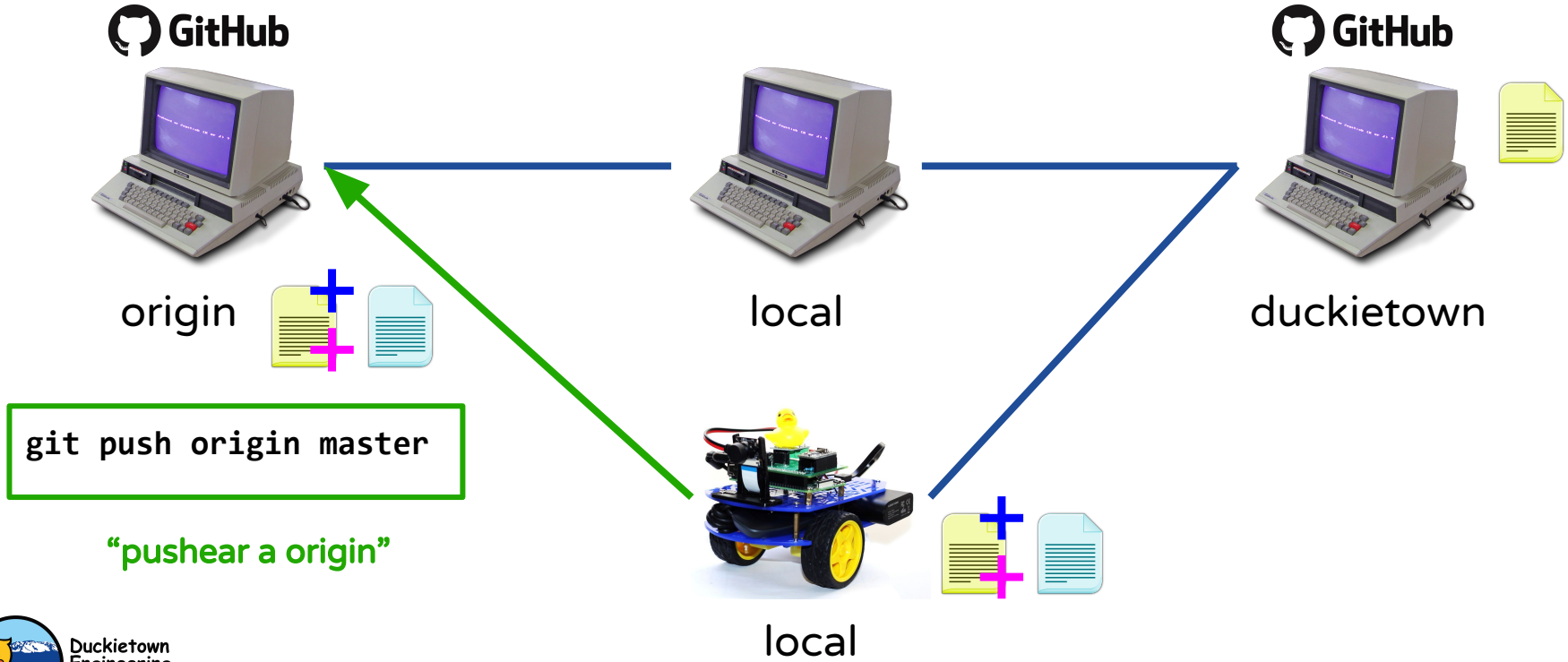


Paso 3

Actualizar la copia del repositorio (fork) de sus cuentas de GitHub con los cambios que hicieron en el robot

a. `git push origin master`

Paso 3



Paso 4

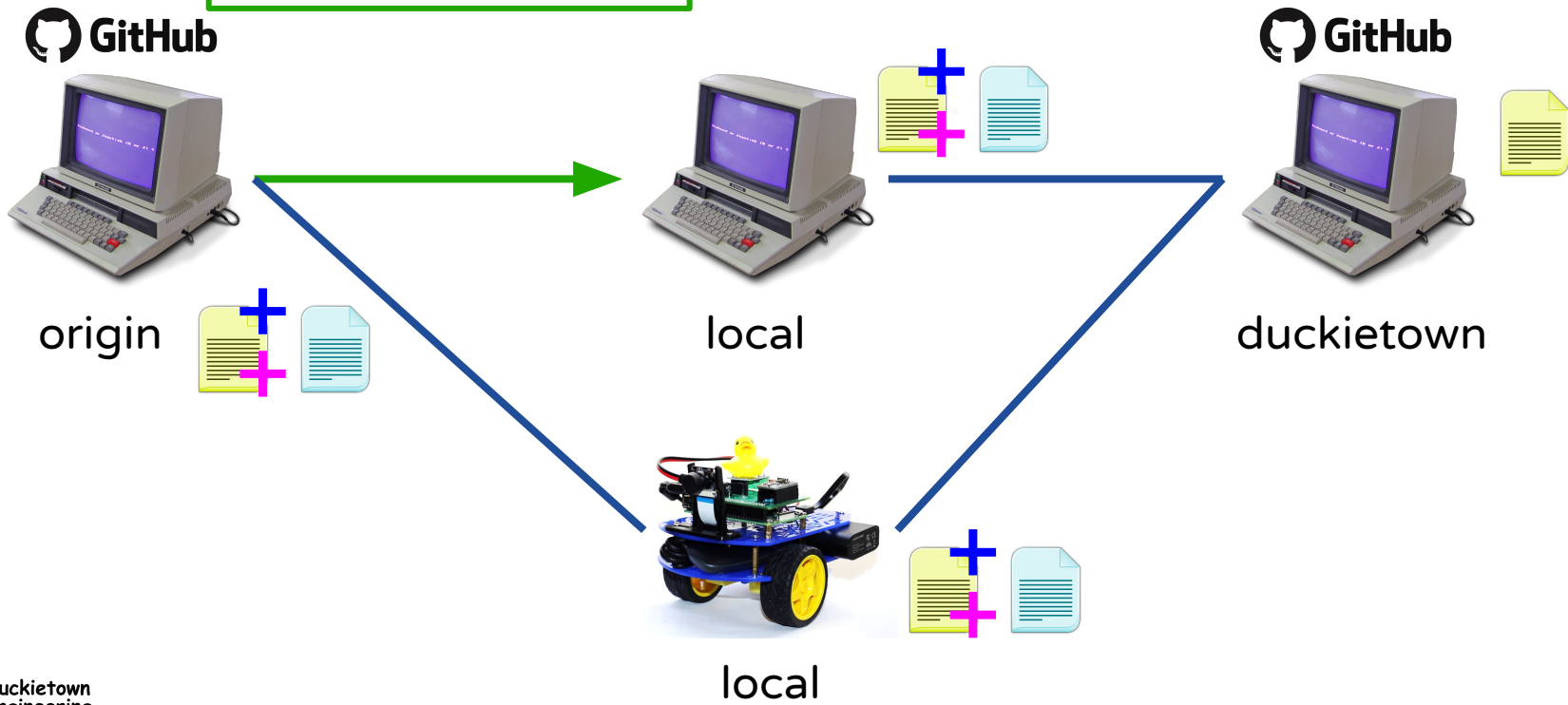
Actualizar la copia de sus laptops con los nuevos cambios del robot

- a. En el laptop: `git pull origin master`

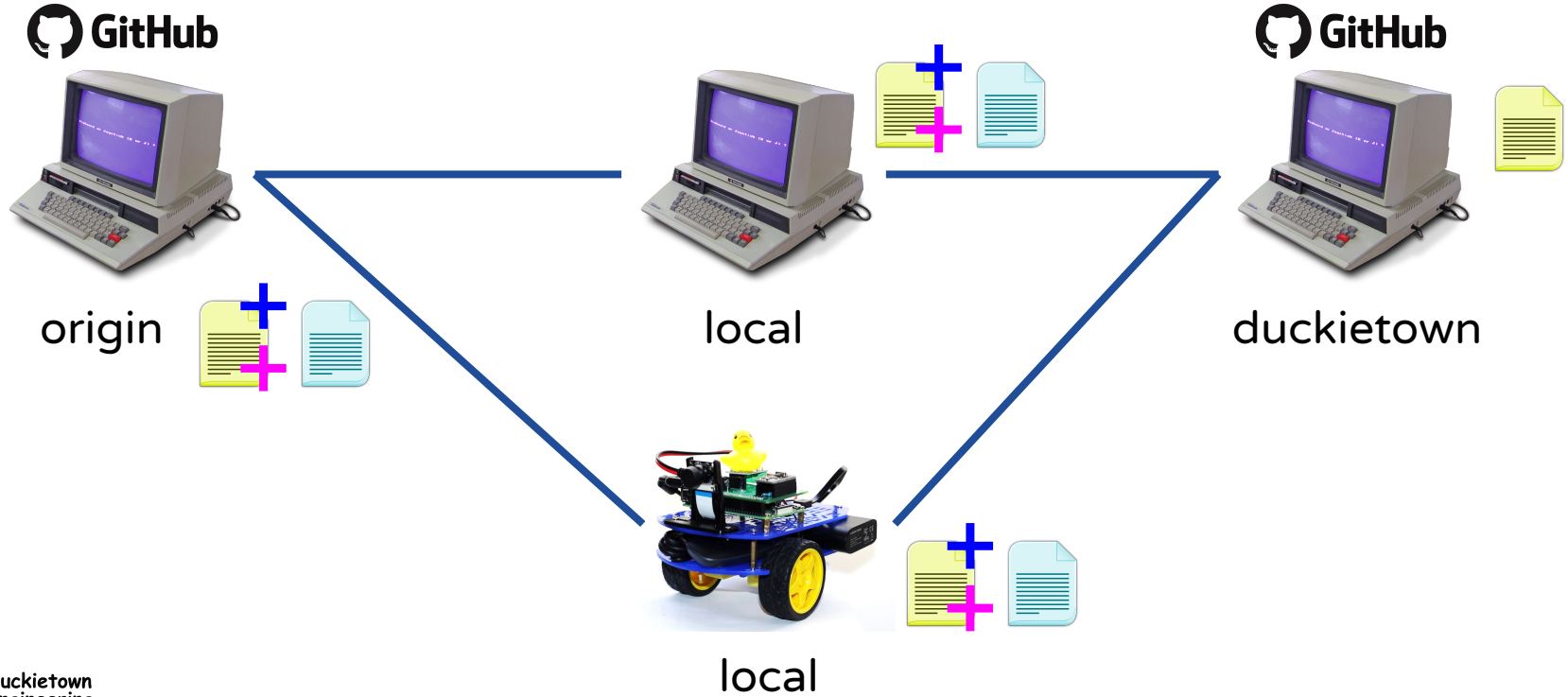
Paso 3

“pulllear origin”

```
git pull origin master
```

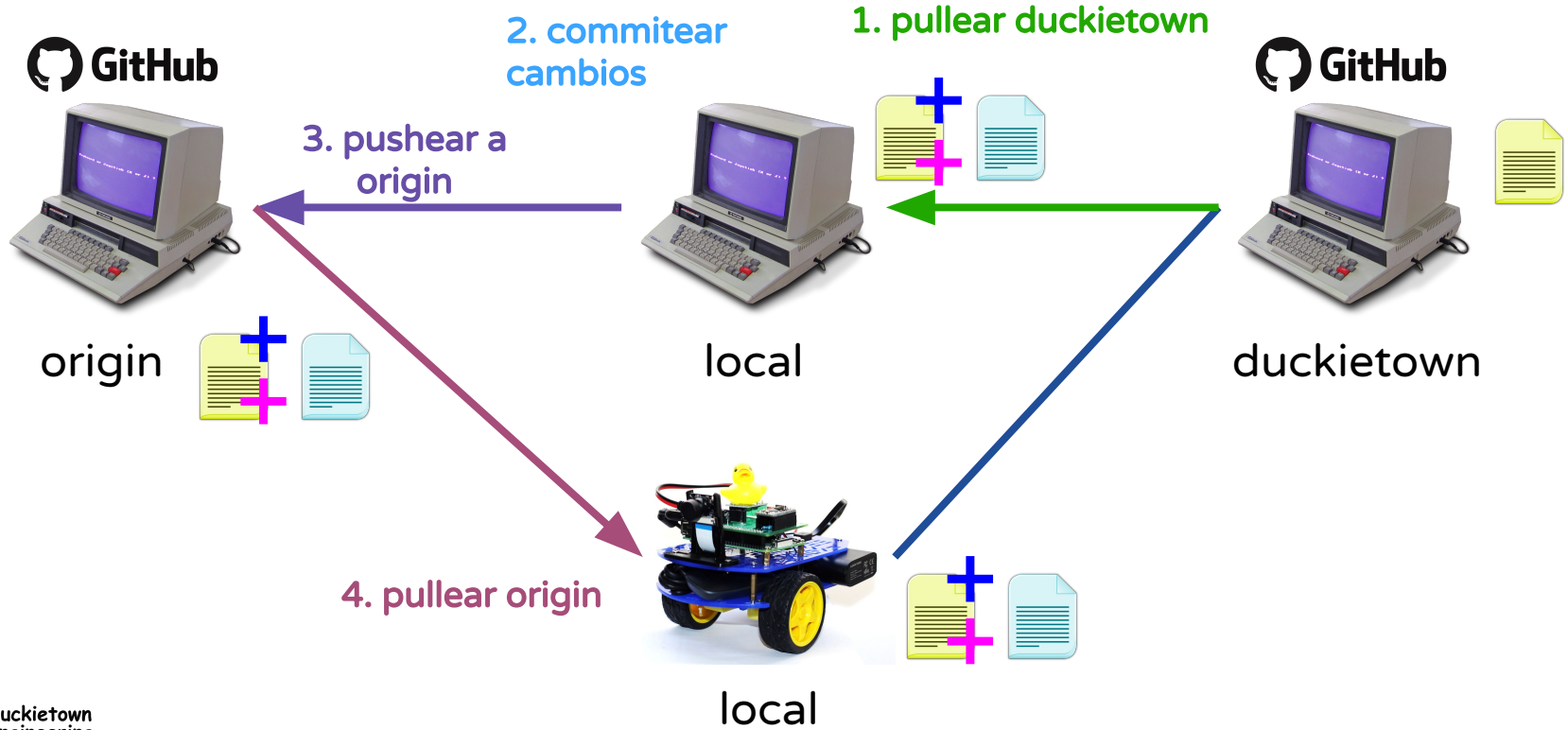


Resultado



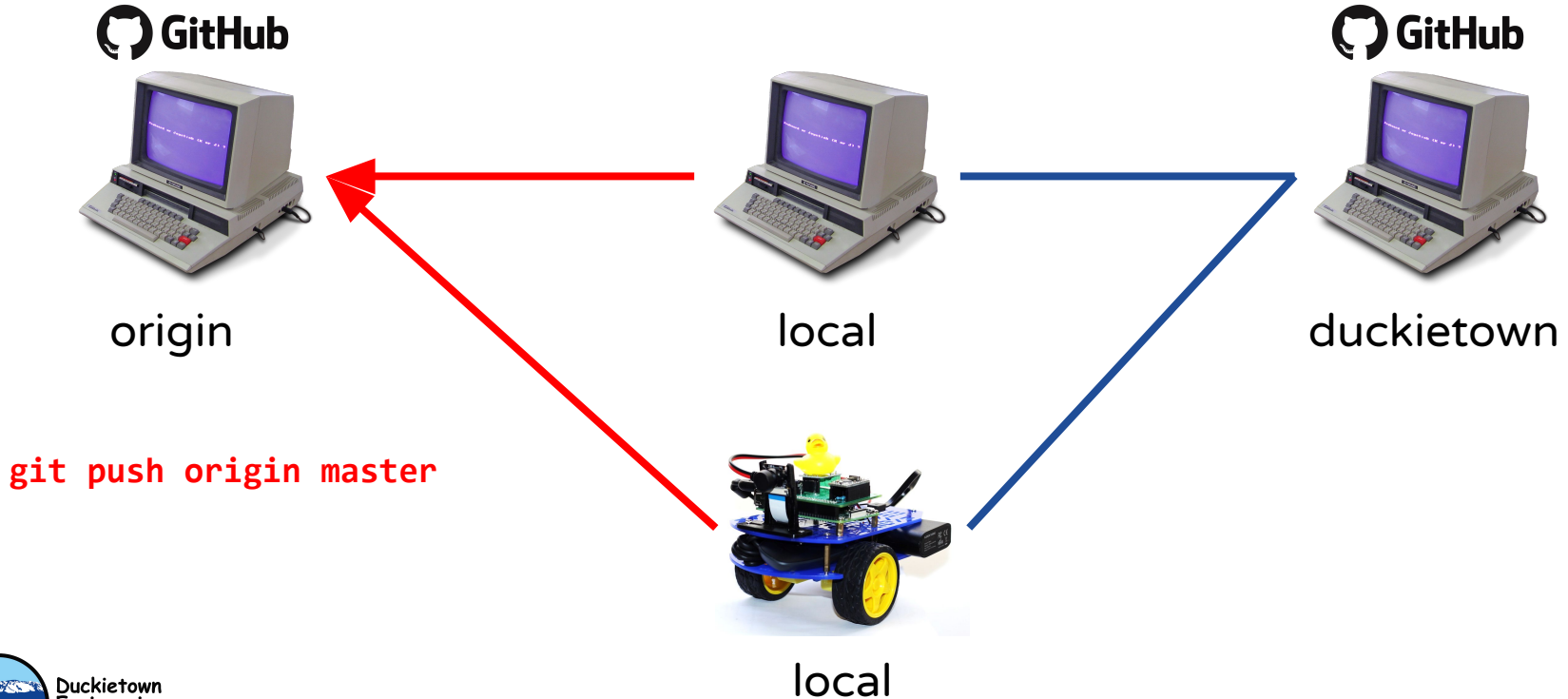
Otros casos

Trabajar en el laptop y actualizar el robot

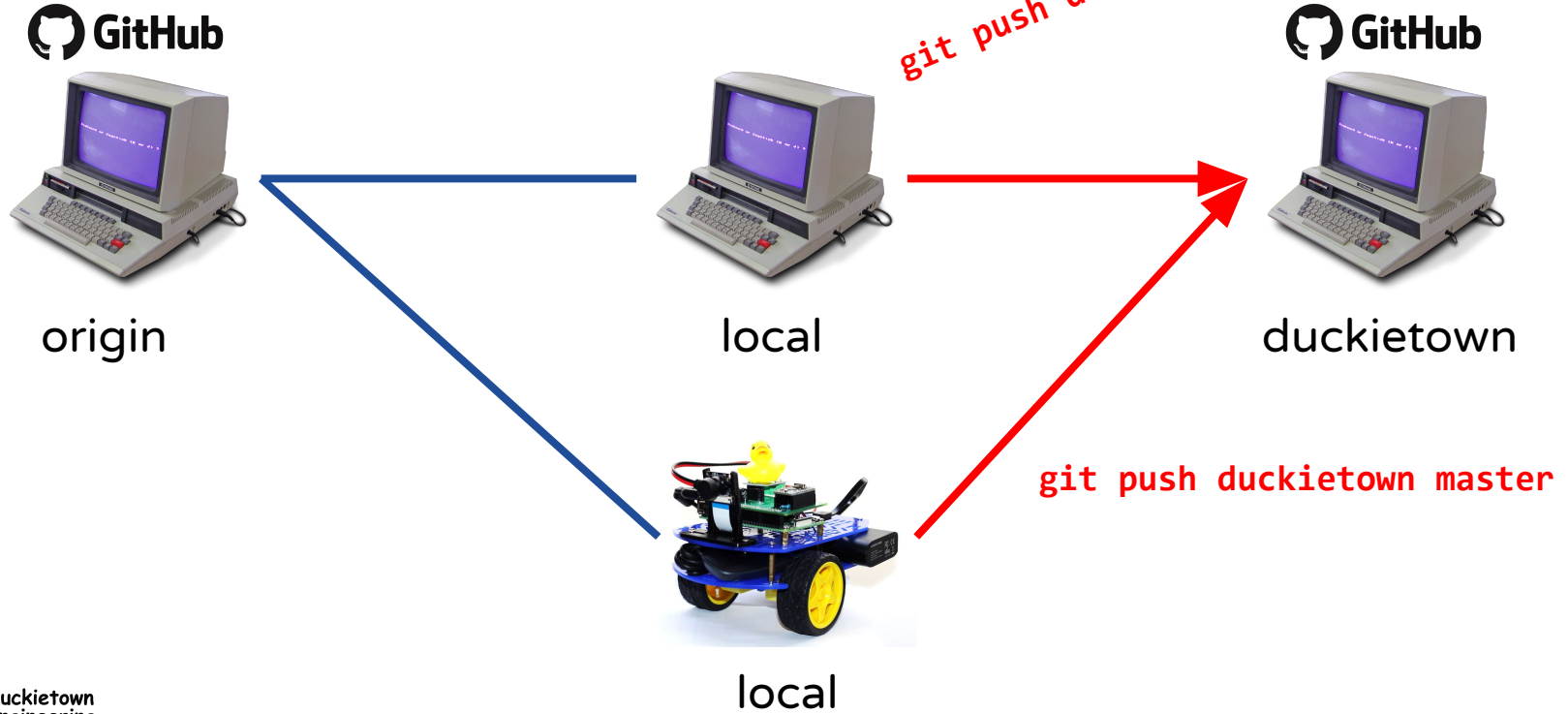


Procedimientos prohibidos

Pushear a origin sin haber pulleado duckietown



Push a duckietown



Actividad práctica de Git

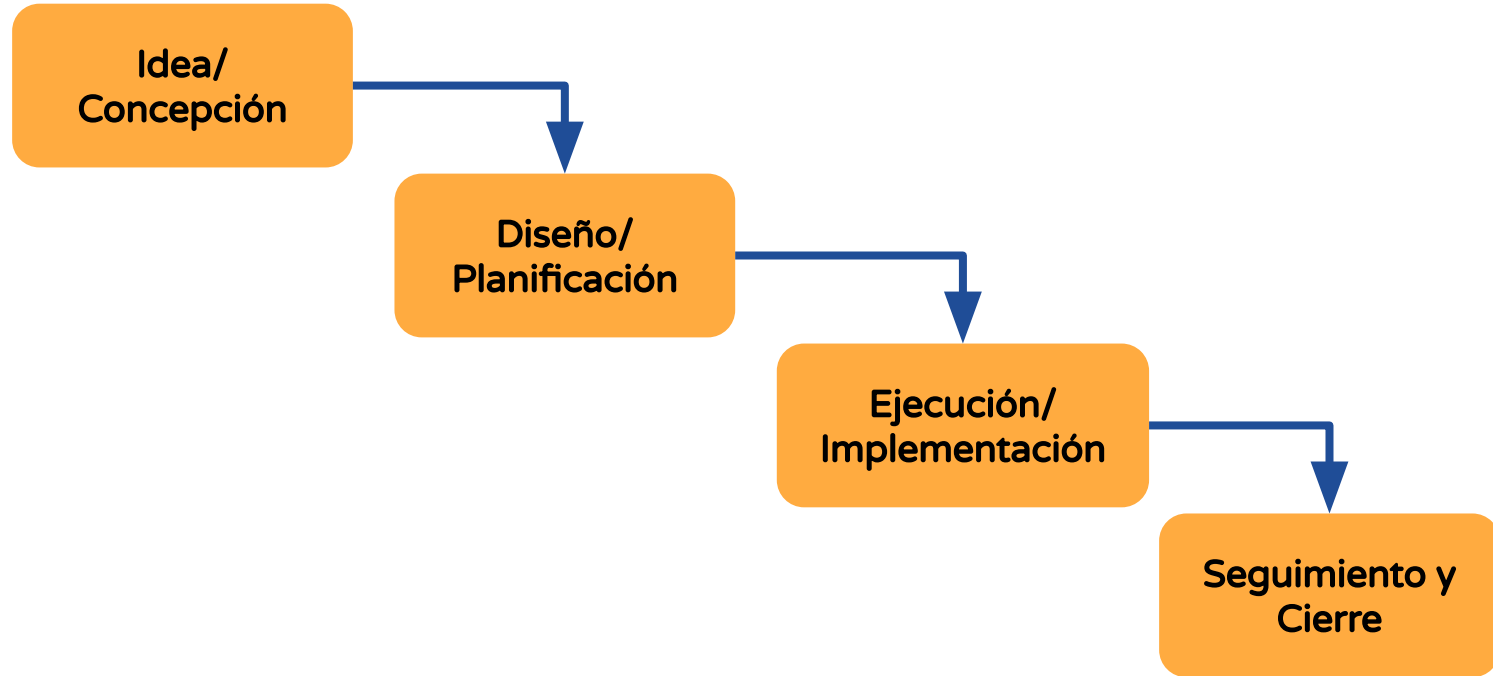
(aburrida pero necesaria)

Test de manejo con logs

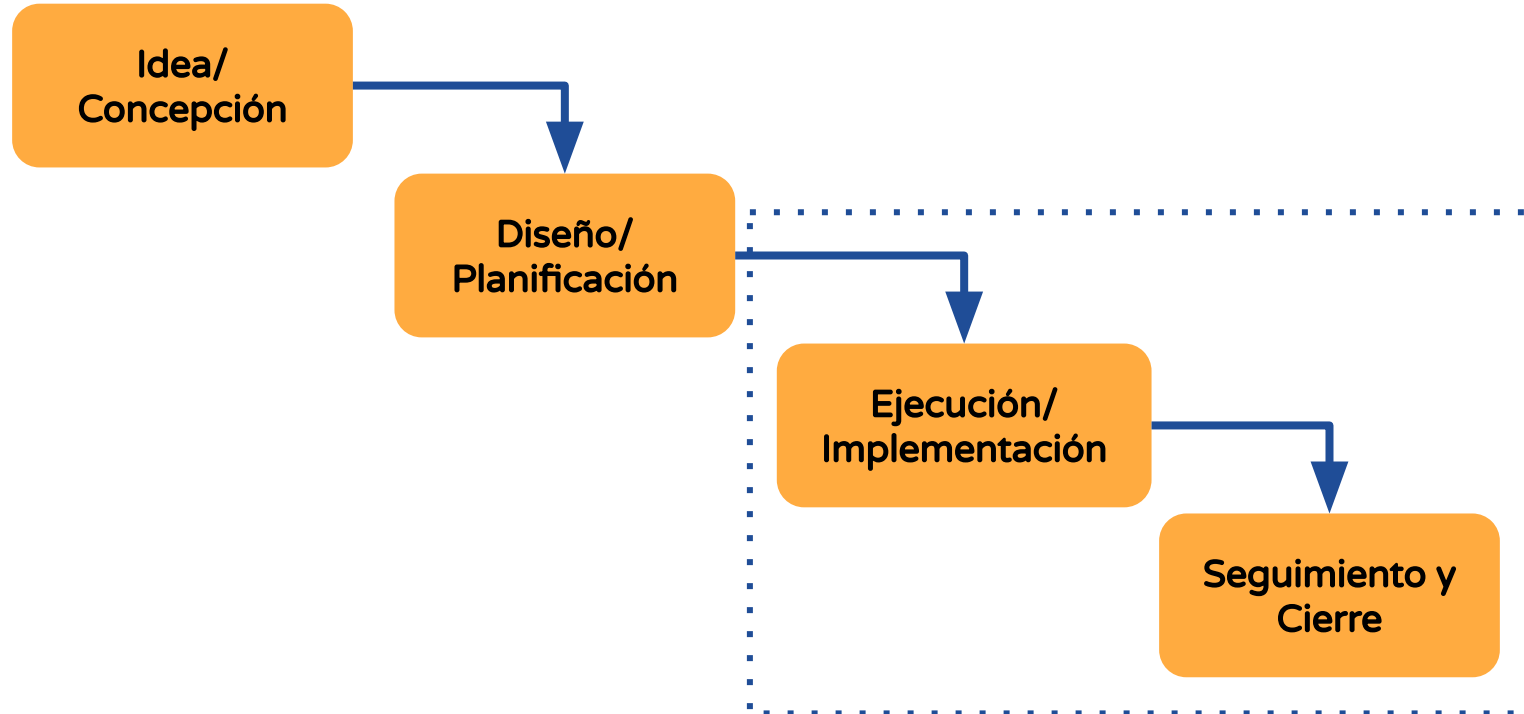
1. Indicar lo que realiza el duckiebot en cada etapa del test de manejo usando los print de ROS
 - a. **loginfo**: avanzar, retroceder, girar
 - b. **logwarn**: freno de emergencia
2. Una vez que hayan comprobado que funcione, deben:
 - a. Commitear los cambios
 - b. Pushearlos a origin
 - c. Pullearlos en sus laptops
 - d. **Observación**: No es necesario *pullear duckietown* en esta oportunidad porque no han habido cambios en el repositorio de la empresa

Anuncios corporativos

Etapas de un proyecto

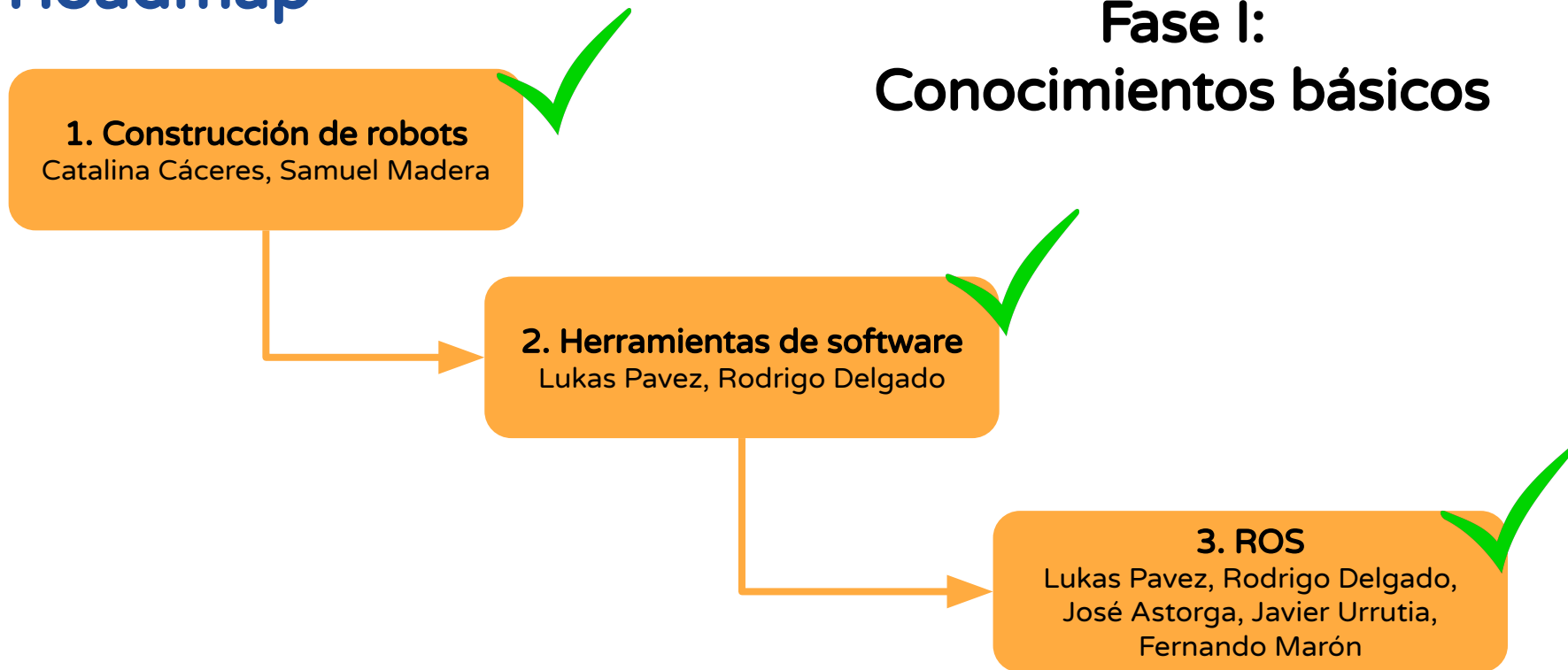


Etapas de un proyecto



Roadmap

Fase I: Conocimientos básicos



Roadmap

Fase II: Conocimientos específicos

4. Visión Computacional
Felipe San Martín



5. Visión 3D
Matías Mattamala (por definir)



6. Autonomía
Por definir

Roadmap

Fase II: Soluciones específicas

