

1.b La estructura de la solución consta de 5 objetos principales: La clase Automóvil, la clase Puente, la Clase Sentido, los Hilos en java y el Semáforo para evitar en el problema de un choque de carros.

Quienes piden los recursos en este caso para poder pasar el puente son los Automóviles que se dividen en clase derecha e izquierda. La Clase Sentido también pide recursos y este es el encargado de cierto tiempo cambiar el sentido en que se pueden transportar los carros del puente.

Quien devuelve los recursos o los libera para cambiar el sentido en que puedan pasar los carros en este caso sería el mismo Puente.

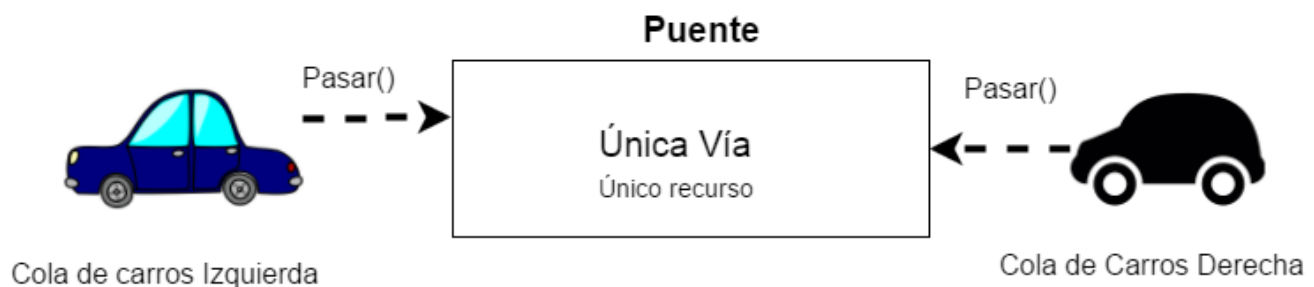
La clase automóvil constará de dos tipos: TipoIzquierda y TipoDerecha, esto nos indicará en qué lado del puente está. Cada vez que lleguen carros de cualquier tipo se añadirán a la cola del tipo correspondiente en la clase Puente.

La clase Sentido tiene la responsabilidad de ser un semáforo en vida real (Distinto a los semáforos de java), es decir que cada cierto tiempo da paso a que los carros del sentido derecho puedan pasar sobre el puente o los del sentido izquierdo puedan pasar el puente, esta es la clase que evitará el problema de la inanición dando paso a los carros de la derecha o izquierda.

La Clase puente es aquella quien tiene la cantidad de carros que están en sus dos lados, así como es el encargado de contar que la cantidad de carros que estén sobre él no sobrepasen a 10 automóviles.

De esta manera con los hilos y los Semáforos de Java para controlar el pedido de recursos (pasar por el puente) simularemos un ambiente real del sistema.

Explicaremos la solución:



Como muestra la gráfica existe el problema de que el puente sólo posee una vía lo que quiere decir que sólo puede dar paso a carros de un sentido a otro pero no los dos al mismo tiempo. Por eso creé la clase Sentido que se encarga de darle paso a los dos lados del puente cada cierto tiempo variable.

Solucionado el problema de dar paso cada cierto tiempo a sólo alguno de los dos lados empezamos entonces a resolver los problemas de sincronización de datos. ¿Qué pasa si cambia el sentido de poder pasar los carros, pero aún sobre el puente existen carros tratando de llegar a su otro lado? ¿Cuándo están pasando los carros de un sentido y llega un carro nuevo para agregarse a la cola? ¿Si llega un carro y el sentido de ese lado está en verde para poder pasar, pero existe una cola de carros delante de

¿? Si no se trata de manejar los recursos para mantener los datos íntegros el programa colapsa con resultados que no simulan un comportamiento en vida real

Por eso se crean 3 Semáforos de Java:

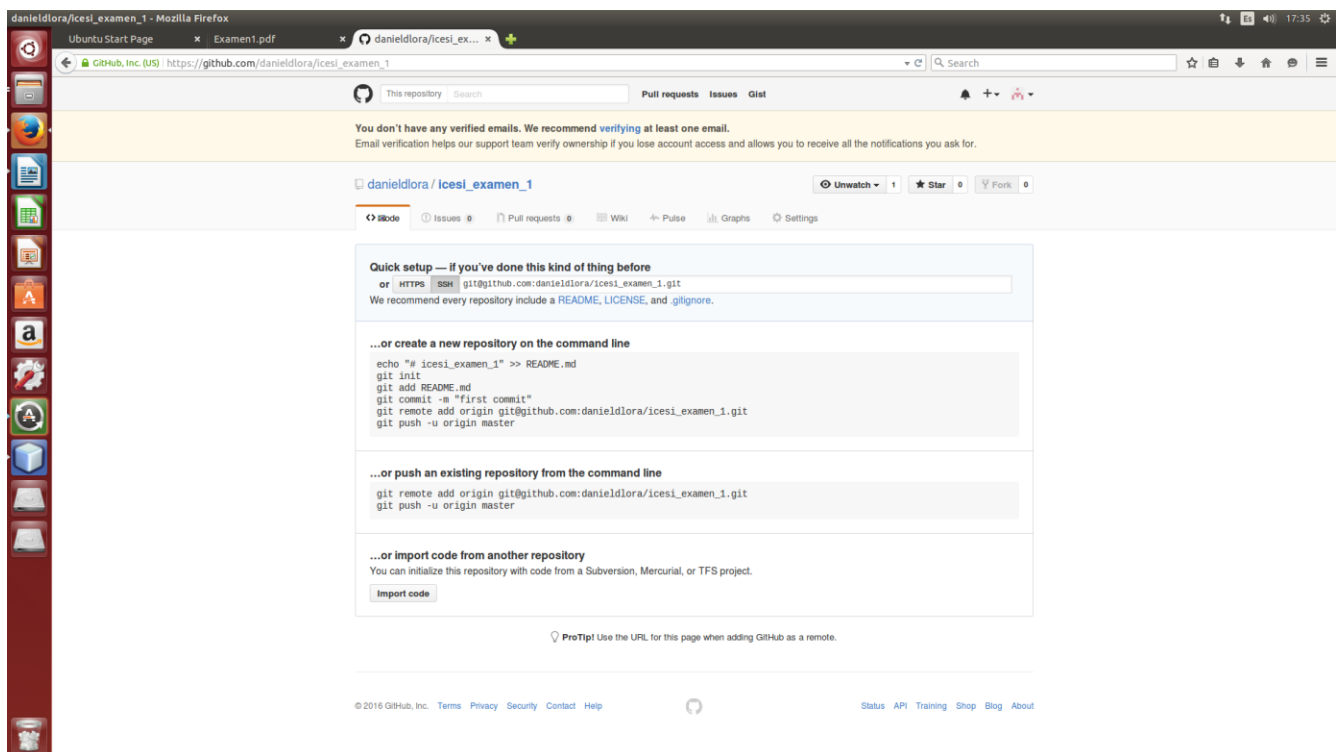
2 son los encargados de asignar los recursos para aumentar la cola de carros en cada lado.

El próximo semáforo está acompañado de una validación por parte de la clase Sentido cambiando en un tiempo indeterminado una variable sentido que posee dos valores: Izquierda y Derecha.

Cuando se valide en qué sentido es que pueden pasar los carros se procede a llamar el método que permite el paso de los carros, pero para mantener la sincronización de procesos de los hilos que llaman al mismo tiempo este método se crea un semáforo que asigna un solo recurso para atender un proceso a la vez y disminuir la cantidad de carros en espera que se da.

Proceso de subida de GIT

El proceso de subida de Github se hizo por SSH



The screenshot displays an IDE environment with the following components:

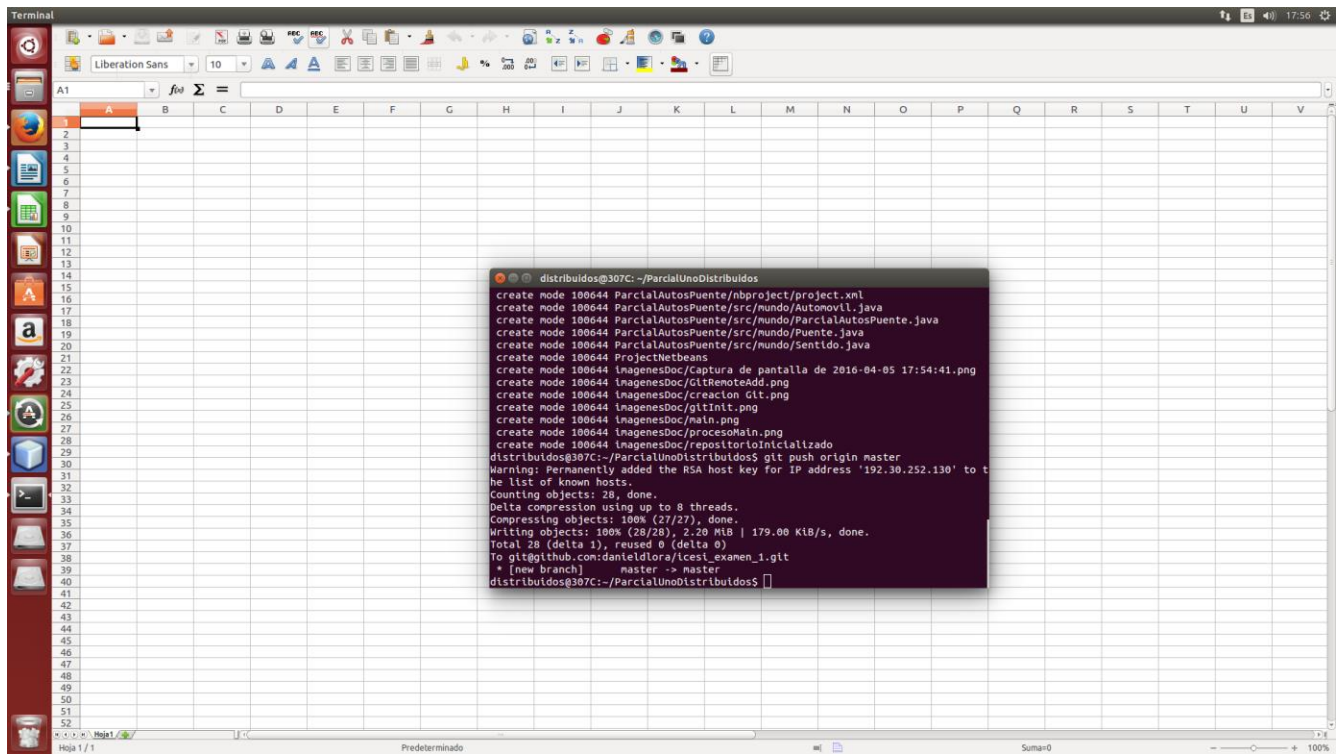
- Project Explorer (Left):** Shows a project named 'Bodega' with sub-packages like 'EjemploHilosSemaforosLectoresE' and 'EjemploHilosSemaforosOficinaC'. The 'Sentido.java' file is selected.
- Code Editor (Center):** Contains the Java code for the 'Sentido' class. The code defines a semaphore with a 'puente' (bridge) and 'sentido' (direction) attributes, and implements a 'run' method that simulates traffic flow.
- Terminal (Foreground):** Shows the execution of 'git' commands in a terminal window. The commands include:
 - `git add --ignore-removal <pathspec>` (which is the current default, ignores paths you removed from your working tree).
 - `git add --all <pathspec>` (will let you also record the removals).
 - `git status` (to check the paths you removed from your working tree).
 - `cd ..` (to move to the parent directory).
 - `cd /home/distribuidos/ParcialUnoDistribuidos` (to navigate to the project directory).
 - `git init` (to initialize a new Git repository).
 - `git add .` (to add all files to the repository).
- Run Console (Bottom):** Shows the output of the program, indicating that the 'Sentido' thread is running.

The image is a screenshot of a web browser displaying the GitHub repository page for 'danielclora/icesi_examen_1'. The browser's address bar shows the URL 'https://github.com/danielclora/icesi_examen_1'. The repository page includes a header with the repository name, a description, and a list of files. A terminal window is overlaid on the page, showing the output of the 'git remote add origin' command, which successfully added the remote origin. The terminal output is as follows:

```
danielclora@icesi_examen_1:~/icesi_examen_1$ git remote add origin git@github.com:danielclora/icesi_examen_1.git
error: Unknown subcommand: origin
usage: git remote [-v | --verbose]
           add [-t <branch>] [-n <master>] [-f] [--tags|--no-tags] [--mirror|--fetch|push] <name> <url>
           rename <old> <new>
           remove <name>
           set-head <name> [-a | --auto | -d | --delete | <branch>]
           show [-n] <name>
           prune [-n | --dry-run] <name>
           update [-p | --prune] [[<group> | <remote>]...]
           set-branches [--add] <name> <branch>...
           set-url [--push] <name> <newurl> [<oldurl>]
           set-url --add <name> <newurl>
           set-url --delete <name> <url>

-v, --verbose          be verbose; must be placed before a subcommand

...or import code from a remote repository
You can initialize this repository with the following command:
git clone https://github.com/danielclora/icesi_examen_1.git
danielclora@icesi_examen_1:~/icesi_examen_1$ git remote add origin git@github.com:danielclora/icesi_examen_1.git
danielclora@icesi_examen_1:~/icesi_examen_1$
```



Link Respositorio: https://github.com/danieldlora/icesi_examen_1