

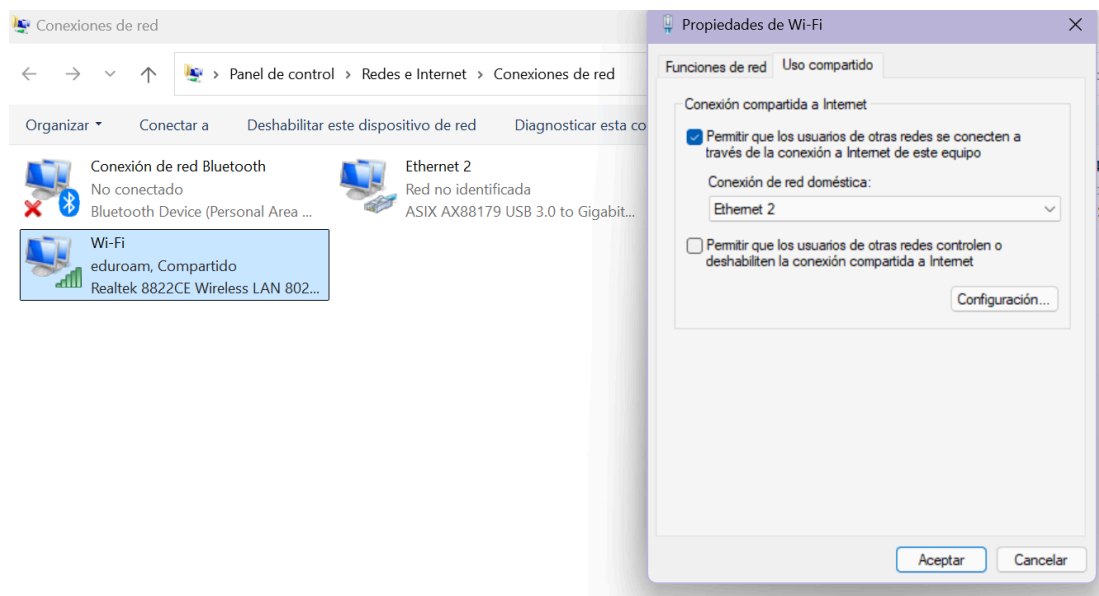
## Memòria - Puzzle 1

Després d'haver configurat la tarjeta MicroSD de la Raspberry Pi 5 amb el sistema operatiu Raspberry Pi OS, amb el nom de host `carla@carla.local`, ja pot ser utilitzada com un ordinador y connectar el perifèric corresponent, en el meu cas el LCD.

La primera opció que vaig provar per connectar amb la Raspberry, va ser per cable Ethernet i un adaptador, donat que el meu portàtil no disposava de port Ethernet.

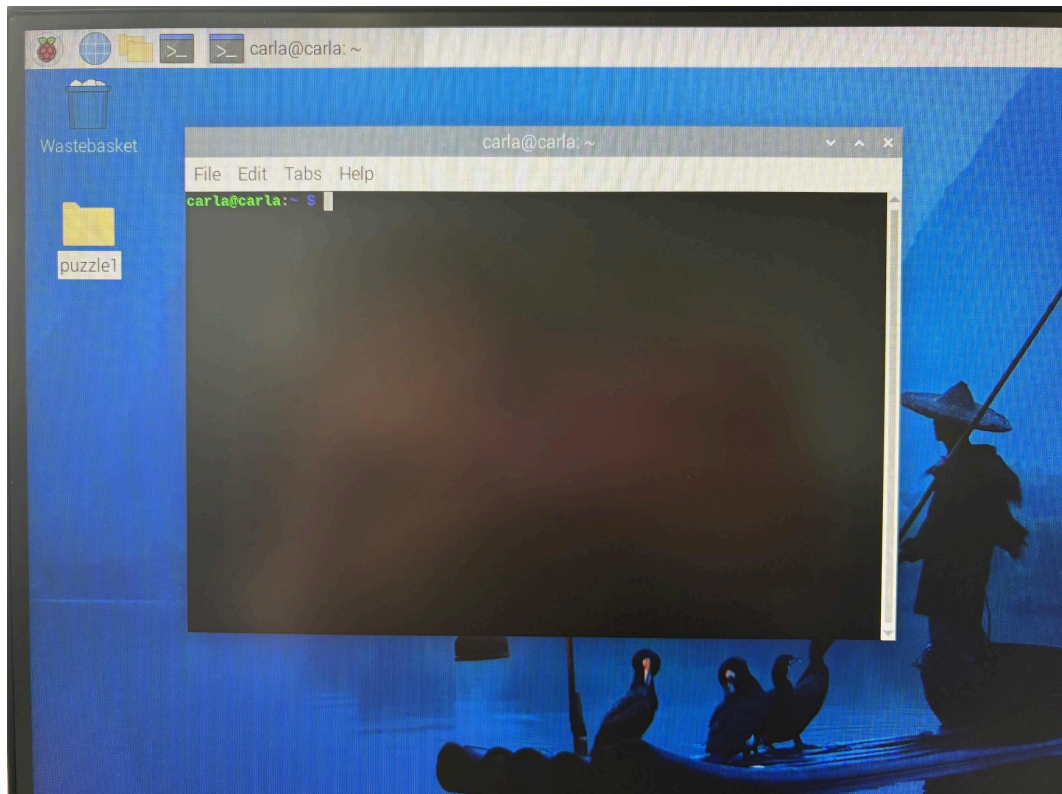
Vaig començar obrint la terminal del meu Windows, i escrivint el comandament `ssh carla@carla.local`, el qual ens permetirà accedir al servidor de la Raspberry, cada vegada que sigui utilitzar. Llavors, per tal d'accedir a la Raspberry i a la seva terminal només havia d'escriure la meva contrasenya escollida. Però que pugui accedir a ella no significa que pugui utilitzar-la completament, ja que com a un ordinador o a un mòbil normal, li ha d'arribar connexió a internet.

Per a això, des del panel de control, s'ha d'habilitar a le spropietats del Wifi del portàtil que aquest pugui ser compartit amb, en aquest cas, l' Ethernet 2 que anava connectat a la meva Raspberry.



Una vegada havent accedit al servidor, vaig utilitzar un altre comandament per comprovar si la connexió proporcionada pel portàtil és correcta, `ping -c 4 google.com`, com no vaig perdre cap dels 4 paquets enviats, sabia que la Raspberry estava connectada a internet.

La meva segona opció, que vaig trobar més fàcil, va ser connectar la Raspberry Pi 5 amb un cable HDMI micro - HDMI a una pantalla de la que disposava. Als ports USB de la Raspberry vaig connectar un teclat i un ratolí. Ja que a la pantalla veia tot el sistema operatiu de l'ordinador, no vaig tenir cap problema en connectar-lo al meu iPhone que gràcies a les dades mòbils em funcionava com a Wi-fi, tenint accés sencer a aquest ordinador.



El següent pas era descarregar Ruby. A la terminal d'aquest s'escriu:

```
$ sudo apt install ruby-full
```

I per comprovar si s'ha instal·lat correctament utilitzarem un comandament que ens mostrarà la versió que tenim:

```
$ ruby -v
```

Ara per tal d'utilitzar el codi font Ruby, per programar un arxiu que faci el que em demana el Puzzle 1, no hi tenia suficient amb les biblioteques que ja venien incloses, i vaig haver de buscar a google, concretament a github, possibles biblioteques relacionades amb el meu I2C a programar.

Buscant un possible codi bàsic per fer display a la pantalla del LCD, vaig trobar un comentari que em recomanava utilitzar la biblioteca o gema *i2c 0.4.2*.

A l'arxiu en el que necessito la biblioteca per poder redactar el codi, havia de inicialitzar els pins del perifèric, i crear les funcions que cridaria més a baix per tal d'enviar el missatge una vegada executat. Però per un problema, que més tard vaig veure que probablement no estava relacionat amb la llibreria, no vaig aconseguir que la pantalla illuminés cap paraula, per tant vaig decidir buscar una altra gema.

Vaig preferir una biblioteca més completa amb la que pogués fer un codi més simple i curt, i la que vaig trobar a Github també estava disponible a RubyGems.org, nomenada *i2c-lcd 0.0.4*.

Els dos cops que vaig instal·lar les biblioteques, vaig escriure a la terminal els comandaments `$ gem search "nom de la gem"` i `$ sudo gem install "nom de la gem"` per instal·lar-les una vegada l'ordinador ja l'hagi trobat amb la versió corresponent.

Ja que a la segona descàrrega incloïa la primera gema que ja tenia, per evitar problemes, vaig desinstal·lar l'*i2c 0.4.2* abans de fer la segona instal·lació ja que l'*i2c-lcd 0.0.4* s'instal·lava com a una mena d'extensió de l'*i2c 0.4.2*.

```
carla@carla:~$ gem search i2c

*** REMOTE GEMS ***

API2Cart (0.0.1)
api2cart-daemon (0.0.1)
api2cart-ruby (1.2.1)
artoo-i2c (0.5.0)
ec2-i2cssh (0.7.1)
i2c (0.4.2)
i2c-alpha_display (0.5.1)
i2c-bme280 (0.1.2)
i2c-devices (0.0.6)
i2c-lcd (0.0.4)
i2c-ss1602 (0.0.2)
i2c2 (0.99.1)
i2cssh (2.2.0)
mruby-linux-i2c (1.0.0)
poi2csv (0.0.5.2)
redi2casa (0.2.13)
ruby-i2c (0.1.0)
ruby_i2c (0.1.3)
```

Amb el comandament `$ gem list`, si s'ha instal·lat bé, trobo la gema a la llista amb les demés tal que:

```
carla@carla:~$ gem list

*** LOCAL GEMS ***

abbrev (default: 0.1.0)
base64 (default: 0.1.1)
benchmark (default: 0.2.0)
bigdecimal (default: 3.1.1)
bundler (default: 2.3.7)
i2c (0.4.2)
i2c-lcd (0.0.4)
```

Ara, el que volia era aprendre a utilitzar la biblioteca en qüestió, ja que no sabia quines funcions i classes incloïa. Vaig estar investigant pels documents penjats al repositori fins que vaig donar amb un apartat nomenat "Usage". Aquest indicava com s'havia d'utilitzar la gem per enviar un missatge a la LCD amb unes 4 línies de codi.

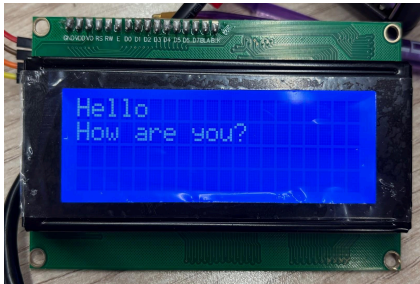
La biblioteca consta d'una classe anomenada Display que és la que conté diferents funcions, com `text`, per poder escriure a la pantalla qualsevol cosa a la línia que vulguem.

```
Usage

require 'i2c/drivers/lcd'
display = I2C::Drivers::LCD::Display.new('/dev/i2c-1', 0x27, rows=20, cols=4)
display.clear
display.text('Hello', 0)
display.text('World', 1)
display.text('Line3', 2)
display.text('Line4', 3)

'rows' and 'cols' options are optional. The default value is 20 and 4.
```

Basant-me en el codi de “Usage” aquest va ser el meu arxiu que cridaré per escriure al LCD, completant així el Puzzle 1:



```
carla@carla: ~/Desktop/puzzle x + v
GNU nano 7.2                                lcd_prueba.rb
require 'i2c/drivers/lcd'
LCD_ADDRESS = 0x27

display = I2C::Drivers::LCD::Display.new('/dev/i2c-1', LCD_ADDRESS, rows=20, cols=4)

display.clear
display.text('Hello', 0)
display.text('How are you?', 1)
```

Finalment, cal esmentar els problemes que he tingut durant aquest exercici. Començant per la connexió de la Raspberry a internet a través del meu portàtil, no acaba d’entendre quines configuracions havia de fer per tal compartir el Wifi al que el portàtil tenia accés, però gràcies als meus companys de grup vaig aconseguir solucionar-ho i entendre-ho per a futures ocasions. Però el major problema que he tingut durant aquestes dues setmanes, ha sigut que vaig confondre els color dels cables que connectaven l’LCD a la Raspberry. Això va conduir a connectar el cable de l’Vcc al Ground i viceversa, pel que segurament la meua LCD està inservible. Gràcies a que un company d’un altre grup em va deixar el seu perifèric, vaig poder comprovar que tant el meu codi com les biblioteques no tenien cap problema i funcionaven a la perfecció.