

# Critical Design Review

## Client

```
Python
import gi
import time
import threading
import requests
from gi.repository import Gtk, GLib, Gdk
from rpi_lcd import LCD
from PUZZLE1_ANNA import RfidPyNFC

gi.require_version('Gtk', '3.0')

class CampusVirtualClient(Gtk.Window):
    def __init__(self):
        super().__init__(title="Campus Virtual")
        self.set_default_size(400, 300)
        self.set_border_width(10)

        self.login_frame = Gtk.Box(orientation=Gtk.Orientation.VERTICAL,
spacing=10)
        self.label_login = Gtk.Label(label="Please, login with your
university card")
        self.label_login.override_background_color(Gtk.StateFlags.NORMAL,
Gdk.RGBA(0, 0, 1, 1)) # Fondo azul
        self.label_login.override_color(Gtk.StateFlags.NORMAL, Gdk.RGBA(1,
1, 1, 1)) # Texto blanco
        self.login_frame.pack_start(self.label_login, True, True, 0)

        self.add(self.login_frame)

        self.nfc_reader = RfidPyNFC()
        self.lcd = LCD()
        self.start_nfc_thread()

    def start_nfc_thread(self):
        self.thread = threading.Thread(target=self.check_nfc)
        self.thread.daemon = True
        self.thread.start()

    def check_nfc(self):
        while True:
```

```

        uid = self.nfc_reader.read_uid()
        if uid:
            Glib.idle_add(self.on_card_detected, uid)
            break

    def on_card_detected(self, uid):
        self.label_login.set_text(f"UID detected: {uid}")
        self.label_login.override_background_color(Gtk.StateFlags.NORMAL,
Gdk.RGBA(0, 1, 0, 1))

        self.load_student_interface(uid)

    def load_student_interface(self, uid):
        self.login_frame.destroy()

        self.student_frame = Gtk.Box(orientation=Gtk.Orientation.VERTICAL,
spacing=10)

        self.lcd.clear()
        self.lcd.text("Welcome", 1)
        self.lcd.text(uid, 2)

        self.option_var = Gtk.ComboBoxText()
        self.option_var.append_text("Timetable")
        self.option_var.append_text("Tasks")
        self.option_var.append_text("Marks")
        self.option_var.set_active(0)
        self.student_frame.pack_start(self.option_var, False, False, 0)

        self.display_button = Gtk.Button(label="Show")
        self.display_button.connect("clicked",
self.on_display_button_clicked)
        self.student_frame.pack_start(self.display_button, False, False, 0)

        self.data_view = Gtk.TextView()
        self.data_buffer = Gtk.TextBuffer()
        self.data_view.set_buffer(self.data_buffer)
        scrolled_window = Gtk.ScrolledWindow()
        scrolled_window.add(self.data_view)
        scrolled_window.set_min_content_height(150)
        self.student_frame.pack_start(scrolled_window, True, True, 0)

        self.add(self.student_frame)
        self.show_all()

    def on_display_button_clicked(self, button):
        option = self.option_var.get_active_text().lower()

```

```

server = f"http://10.192.91.240:8000/{option}"
try:
    response = requests.get(server)
    response.raise_for_status()
    data = response.text
    self.display_data(data)

    self.lcd.clear()
    self.lcd.text(option.capitalize(), 1)
    lines = data.split('\n')[:3]
    for i, line in enumerate(lines):
        self.lcd.text(line[:20], i + 2)
except requests.RequestException as e:
    self.display_data(f"Error loading {option}:\n{e}")

def display_data(self, data):
    self.data_buffer.set_text(data)

if __name__ == "__main__":
    app = CampusVirtualClient()
    app.connect("destroy", Gtk.main_quit)
    app.show_all()
    Gtk.main()

```

Els imports són bastant semblants que el puzzle 2 del Elechouse, ja que necessitem executar processos en segon pla i necessitem les llibreries necessàries per construir l'entorn gràfic.

Hem trobat interessant fer un menú desplegable amb les diferents opcions de taula que pot triar l'usuari i al fer "clic" en l'opció desitjada, ja apareix la taula.

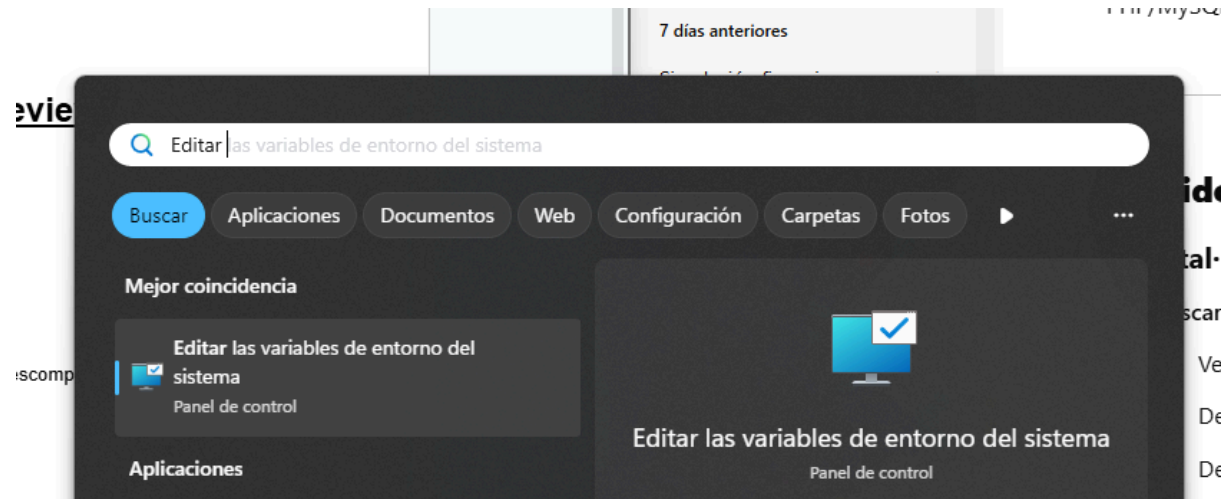
Ara mateix tenim el problema que les dades rebudes del servidor estan en format JSON i estem intentant modificar el codi per convertir el format en l'adequat.

També tenim una mica de dificultat a l'hora de la connexió, ja que hem d'estar tant el client com el servidor, connectats al mateix wifi i no serveix l'eduroam, ja que aquest té un modem personal per compte, un proxy diferent per persona i no ens permet fer la connexió servidor-client.

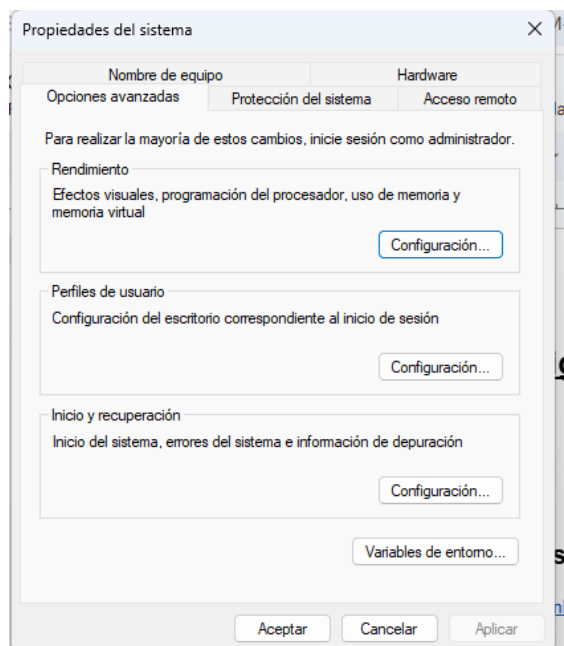
## Servidor (Des de un Windows)

Descargar PHP (<https://windows.php.net/download/>) i descomprimir-lo a C:\php

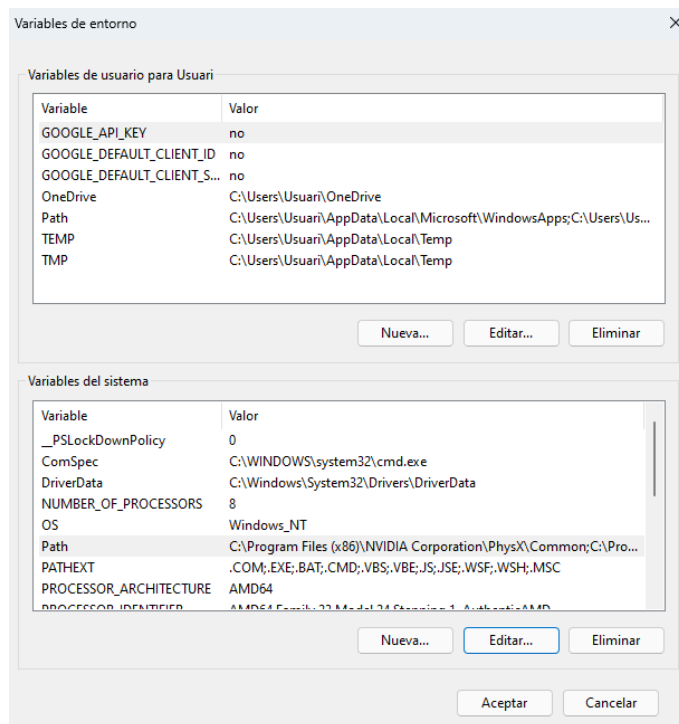
Configurar el path a les variables d'entorn del sistema:



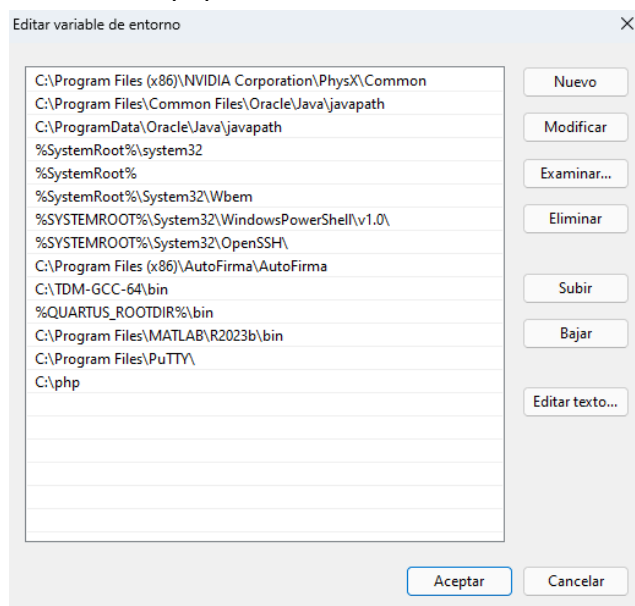
Entrar a “Editar las variables de entorno del sistema” i clicar a “Variables de entorno”:



Editam les “Variables del sistema”:



Escribim "C:\php":



Configurar el PHP: Obrir un terminal (php -ini), buscar a la carpeta de php el php.ini (en el meu cas php.ini-development, llavors fem una copia i li canviem el nom a php.ini), dins aquest document descomentar l'extensió "mysqli" (extension = mysqli) i guardar.

Descarrega MySQL (<https://dev.mysql.com/downloads/mysql/>) i descomprimir-lo a C:\mysql

Configurar el MySQL, creant el fitxer my.ini:

```
Unset
[mysqld]
basedir=C:/mysql           //Directori base de MySQL
datadir=C:/mysql/data      //Directori on es guarden les bases de dades
port=3306                  //Port del MySQL
sql_mode=NO_ENGINE_SUBSTITUTION, STRICT_TRANS_TABLES
```

Obrir el terminal (com a admin), entrar dins de la carpeta bin (C:\mysql\bin) i inicialitzar la base de dades (mysqld --initialize-insecure --basedir=C:\mysql --datadir=C:\mysql\data)

Ara iniciem la connexió amb ambdós serveis:

### MySQL:

Obrir la CMD:

- cd C:\mysql\bin
- mysqld --console
- (Deixem la consola oberta)

Obrir una altra pestanya de la CMD:

- cd C:\mysql\bin
- mysql -u root (en cas de haver ficat una contrasenya "-p" al final, i després del "ENTER" escriure la contrasenya)
- Llavors ja estarem dins de l'entorn de mysql, aquí podrem interactuar amb la base de dades.

### PHP:

Obrir una CMD (una altra pestanya):

- cd C:\projecte (Carpeta on estigui el document .php que volem executar)
- php -S ip\_wifi:port (En cas de voler crearlo de forma local la ip seria localhost)

Per descobrir la ip:

- Obrir una altra CMD (una altra pestanya)
- ipconfig
- Buscar la IP dins de l'apartat de la Wi-Fi o la red que es volen connectar amb el client i buscar la IPv4

```

C/C++
<?php
$servername = "localhost"; //Configuracion inicial
$username = "root";
$password = "1234";
$dbname = "atenea";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    http_response_code(500); //Error de servidor - Conexion Fallida
    die(json_encode(["error" => "Conexion fallida: " . $conn->connect_error]));
}

session_start(); // Inicio de sesion
if (isset($_SESSION['last_activity']) && (time() - $_SESSION['last_activity'] > 900))
{ //15 minutos de inactividad
    session_unset();
    session_destroy();
    http_response_code(401); //Error de cliente - Tiempo Maximo de inactividad
    superado
    die(json_encode(["error" => "Sesion expirada."]));
}
$_SESSION['last_activity'] = time();

$uid = $_GET['uid'] ?? null; //Validar UID
if (!$uid) {
    http_response_code(400); //Error de cliente - UID no encontrada
    die(json_encode(["error" => "UID no proporcionado."]));
}

$student_check = $conn->query("SELECT name FROM students WHERE uid = '" .
    $conn->real_escape_string($uid) . "'");
if ($student_check->num_rows === 0) {
    http_response_code(404); //Error de cliente - UID no registrada
    die(json_encode(["error" => "UID no encontrado en la base de datos."]));
}
$student_name = $student_check->fetch_assoc()['name'];
echo "Bienvenido, $student_name.\n";

$query = $_SERVER['QUERY_STRING'] ?? null; //Comprobar si QUERY_STRING esta definida

$request_uri = $_SERVER['REQUEST_URI'] ?? ''; //Separar la tabla de las
restricciones
$request_parts = explode('?', $request_uri, 2); //Partir el string central con el
'?'
$table = ltrim($request_parts[0], '/'); //La primera parte es la tabla

$allowed_tables = ['marks', 'tasks', 'timetables']; //Validar tabla
if (!$table || !in_array($table, $allowed_tables)) {
    http_response_code(400); //Error de Cliente - Tabla Introducida
    Invalida
    die(json_encode(["error" => "Tabla no valida."]));
}

```

```

$params = []; //Analizar las restricciones si existen
if (!empty($request_parts[1])) { //La segunda parte son las restricciones
    parse_str($request_parts[1], $params); //Dividimos las restricciones
}

$operator_map = [ //Mapeo de operadores soportados
    'lt' => '<',
    'lte' => '<=',
    'gt' => '>',
    'gte' => '>=',
    'eq' => '=',
];

$constraints = []; //Procesar restricciones
$order_by = "";
$limit = "";

foreach ($params as $column => $conditions) { //Análisis de cada condición o
    restricción
    if (is_array($conditions)) { //Separar las restricciones directas
        foreach ($conditions as $operator => $value) {
            if (isset($operator_map[$operator])) {
                if ($value === 'now') { //Manejar valores especiales como 'now'
                    $value = (strpos($column, 'date') !== false) ? date('Y-m-d') :
date('H:i');
                }
                $constraints[] = $column . " " . $operator_map[$operator] . " '" .
$conn->real_escape_string($value) . "'";
            }
        }
    } else {
        $constraints[] = $column . " = '" . $conn->real_escape_string($conditions) .
"'"; //Restricción directa
    }
}

if (isset($params['limit']) && is_numeric($params['limit'])) { //Procesar el
    parámetro limit
    $limit = "LIMIT " . intval($params['limit']);
}

switch ($table) { //Ordenamiento por defecto (marks en subjects, tasks por date i
    timetables por día/hora)
    case 'marks':
        $order_by = "ORDER BY subject";
        break;
    case 'tasks':
        $order_by = "ORDER BY date";
        break;
    case 'timetables': //Agregar la ordenación correcta por día de la semana
        i hora

```



```

        $order_by = "ORDER BY
            CASE day
                WHEN 'Mon' THEN 1
                WHEN 'Tue' THEN 2
                WHEN 'Wed' THEN 3
                WHEN 'Thu' THEN 4
                WHEN 'Fri' THEN 5
                WHEN 'Sat' THEN 6
                WHEN 'Sun' THEN 7
                ELSE 8
            END, hour"; //Primero ordena por dia, luego por hora
        break;
    }

    if (isset($params['limit']) && is_numeric($params['limit'])) { //Procesar el
    parametro limit
        $limit = "LIMIT " . intval($params['limit']);
    }

    $sql = "SELECT * FROM $table"; //Construir la consulta SQL

    if (!empty($constraints)) { //Comprobar si hay restricciones y procesarlas
        if($table !== 'marks'){ //Si no es la tabla 'marks'
            $constraints = array_filter($constraints, function($constraint) {
                return !str_contains($constraint, 'uid'); //Eliminar 'uid' de las
    restricciones
            });
        }

        $constraints = array_filter($constraints, function($constraint) {
            return !str_contains($constraint, 'limit'); //Eliminar 'limit' de las
    restricciones
        });

        if (!empty($constraints)) {
            $sql .= " WHERE " . implode(" AND ", $constraints); //Si hay otras
    restricciones, agregarlas
        }
    }

    $sql .= " $order_by"; //Ordenar como es pedido

    $sql .= " $limit"; //Poner el limite (si se necesita)

    //echo "SQL Query: " . $sql . "\n"; //Verificar codigo SQL

    $result = $conn->query($sql); //Ejecutar la codigo SQL

    if (!$result) {
        http_response_code(501); //Error de servidor - Mensaje Fallido
        die(json_encode(["error" => "Consulta SQL fallida: " . $conn->error]));
    }

```

```

$data = [];
while ($row = $result->fetch_assoc()) {    //Recoger resultados
    $data[] = $row;
}

header('Content-Type: application/json');    //Resultados en JaSON Derulo
echo json_encode($data, JSON_PRETTY_PRINT);

$conn->close();
?>

```

## Base de dades

La base de dades organitza tota la informació del sistema: estudiants, tasques, horaris i qualificacions, vinculant-la amb un UID únic per identificar cada usuari.

El codi a MySQL necessari per a la creació i configuració de la base de dades es detalla a continuació:

```

-- Crear la base de datos
CREATE DATABASE IF NOT EXISTS atenea;
USE atenea;

-- Creación de las tabla que usaremos
CREATE TABLE IF NOT EXISTS students(
    uid VARCHAR(8) NOT NULL,
    name VARCHAR(30),
    PRIMARY KEY(uid)
) ENGINE = INNODB;

CREATE TABLE IF NOT EXISTS tasks(
    date DATE NOT NULL,
    subject VARCHAR(30),
    name VARCHAR(30)
) ENGINE = INNODB;

CREATE TABLE IF NOT EXISTS timetables(
    day VARCHAR(3),
    hour INT,
    subject VARCHAR(30) NOT NULL,
    room VARCHAR(7)
) ENGINE = INNODB;

CREATE TABLE IF NOT EXISTS marks(
    subject VARCHAR(10),
    name VARCHAR(30),
    mark INT,
    uid VARCHAR(8) NOT NULL,
    CONSTRAINT fk_id
    FOREIGN KEY(uid)
    REFERENCES students(uid)
) ENGINE = INNODB;

```

```

-- Cambiar el método de autenticación del usuario root y establecer contraseña
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1234';
-- Aplicar los cambios de permisos
FLUSH PRIVILEGES;

-- Insertar datos de estudiantes
INSERT INTO students(uid, name)
VALUES
    ('A2198D27', 'Eloi Saballs'),
    ('169C9314', 'Èric Lozano'),
    ('13503125', 'Marc Muñoz'),
    ('66269414', 'Anna Llamas'),
    ('06F3E0B0', 'Alejandro de Alvarado');

-- Insertar datos de tareas
INSERT INTO tasks (date, subject, name)
VALUES
    ('2024-12-10', 'PBE', 'Puzzle 1'),
    ('2024-12-15', 'PSAVC', 'Practica 2'),
    ('2024-12-17', 'RP', 'Practica 3'),
    ('2024-12-22', 'TD', 'Entrega 2'),
    ('2024-12-28', 'DSBM', 'Practica 3'),
    ('2024-12-30', 'PBE', 'Critical Design Report');

-- Insertar datos del horario
INSERT INTO timetables(day, hour, subject, room)
VALUES
    ('Mon', 10, 'RP', 'A4105'),
    ('Mon', 16, 'TD', 'A4105'),
    ('Tue', 10, 'DSBM', 'A4105'),
    ('Tue', 12, 'LAB DSBM', 'C5S101A'),
    ('Tue', 16, 'PSAVC', 'A4105'),
    ('Wed', 08, 'PBE', 'A4105'),
    ('Wed', 17, 'LAB RP', 'D3006'),
    ('Thu', 08, 'PBE', 'A4105'),
    ('Thu', 10, 'RP', 'A4105'),
    ('Thu', 14, 'TD', 'A4105'),
    ('Fri', 12, 'DSBM', 'A4105'),
    ('Fri', 15, 'PSAVC', 'A4105');

-- Insertar calificaciones para cada estudiante
INSERT INTO marks(subject, name, mark, uid)
VALUES
    ('EIM', 'Examen Parcial', 8.4, 'A2198D27'),
    ('TCGI', 'Examen Final', 7.1, 'A2198D27'),
    ('PBE', 'CDR', 8.6, 'A2198D27'),
    ('PSAVC', 'Examen Final', 5.6, 'A2198D27'),
    ('RP', 'Examen Lab', 9.3, 'A2198D27');

INSERT INTO marks(subject, name, mark, uid)
VALUES
    ('CAL', 'Examen Parcial', 1.5, '169C9314'),
    ('ALG', 'Examen Parcial', 0.2, '169C9314'),
    ('FDF', 'Examen Parcial', 5.1, '169C9314'),
    ('FDE', 'Examen Parcial', 7.5, '169C9314'),
    ('FO', 'Examen Parcial', 9.7, '169C9314'),
    ('TFG', 'Entrega Final', 10.0, '169C9314');

INSERT INTO marks(subject, name, mark, uid)
VALUES
    ('CAL', 'Examen Final', 9.2, '13503125'),

```

```
('EM', 'Examen Parcial', 3.1, '13503125'),  
('FISE', 'Examen Parcial', 7.5, '13503125'),  
('RP', 'Examen Parcial', 4.3, '13503125'),  
('EIM', 'Examen Final', 9.6, '13503125');
```

```
INSERT INTO marks(subject, name, mark, uid)  
VALUES  
('FDE', 'Examen Parcial', 8.2, '66269414'),  
('ALG', 'Examen Final', 3.4, '66269414'),  
('FO', 'Examen Final', 10.0, '66269414'),  
('ENTIC', 'Projecte Final', 9.1, '66269414'),  
('POO', 'Projecte Final', 6.5, '66269414');
```

```
INSERT INTO marks(subject, name, mark, uid)  
VALUES  
('POO', 'Projecte', 9.9, '06F3E0B0'),  
('PIE', 'Examen Final', 4.7, '06F3E0B0'),  
('IPAV', 'Examen Final', 9.7, '06F3E0B0'),  
('ICOM', 'Examen Parcial', 6.5, '06F3E0B0'),  
('DSBM', 'Examen Lab', 1.8, '06F3E0B0');
```