

Critical Design Review

(CDR)

Grup: Ivan Rodríguez, Junle Wang, Mariona Capdevila, Blanca Calero i Roger Garcia

Índice

1. Inicialización.....	2
2. Servidor:.....	3
3. Cliente.....	7
4. Database.....	14
5. Conexiones.....	17
5.1. Conexión cliente - servidor.....	17
5.2. Conexión servidor - database.....	18
6. Cliente web.....	19

Para esta entrega de CDR, hemos decidido mejorar el anterior, siendo más claros y mejorando el código para que hiciera lo necesario. Además hemos implementado un cliente **web para pedir los datos desde diferentes ordenadores**.

Todo el código del proyecto se puede encontrar en: [GITHUB CODE](#)

1. Inicialización

Si trabajamos en Windows lo que podemos hacer es instalarnos un terminal linux que nos permitirá trabajar con ubuntu. Abrimos la terminal de windows y lo que tenemos que hacer es poner el comando de:

wsl –install Ubuntu-24.04 (ojo que son dos guiones , si trabajas con ubuntu no hace falta este paso)

```
Ubuntu ya está instalado.
Iniciando Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: junle
New password:
Retype new password:
```

Introducimos un nombre de usuario que queramos usar, y ponemos contraseña.

PHP

A Continuación lo que haremos es añadir el repositorio Ondřej Surý, que es un repositorio que ofrece versiones más nuevas de PHP en Ubuntu.

```
sudo apt update
sudo apt install -y software-properties-common
sudo add-apt-repository ppa:ondrej/php
```

Nos dirá que si queremos continuar le damos al **ENTER** o si queremos cancelar **ctrl+c**. Le daremos a continuar presionando el **ENTER**.

Actualizaremos la lista de paquetes disponibles.

```
sudo apt update
```

Descargamos **PHP versión 8.3** que es la que hemos empezado a utilizar, php con un soporte en mysql.

```
sudo apt install php8.3-fpm
sudo apt install php8.3-mysql
```

Pondremos el comando de `php -v` para comprobar que la versión que hemos descargado es la correcta.

```
junle@JuninhoPC:~$ php -v
PHP 8.3.14 (cli) (built: Nov 25 2024 18:07:43) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.14, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.14, Copyright (c), by Zend Technologies
```

Comandos importantes de php

sudo systemctl enable php8.3-fpm

```
ivan_rodriguez :: verty002-pc → /var/www/html » sudo systemctl enable php8.3-fpm
Synchronizing state of php8.3-fpm.service with SysV service script with /usr/lib/systemd/systemd-sysv-...
Executing: /usr/lib/systemd/systemd-sysv-install enable php8.3-fpm
ivan_rodriguez :: verty002-pc → /var/www/html »
```

Activa el php-fpm

sudo systemctl status php8.3-fpm

```
ivan_rodriguez :: verty002-pc → - » sudo systemctl status php8.3-fpm
[sudo] contraseña para ivan_rodriguez:
● php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager
  Loaded: loaded (/usr/lib/systemd/system/php8.3-fpm.service; enabled; preset: enabled)
  Active: active (running) since Thu 2024-12-12 17:05:56 CET; 16min ago
    Docs: man:php-fpm8.3(8)
   Process: 1621 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run...
  Main PID: 1594 (php-fpm8.3)
    Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req>
     Tasks: 3 (limit: 18401)
    Memory: 15.1M (peak: 16.1M)
      CPU: 119ms
     CGroup: /system.slice/php8.3-fpm.service
             ├─1594 "php-fpm: master process (/etc/php/8.3/fpm/php-fpm.conf)"
             ├─1619 "php-fpm: pool www"
             └─1620 "php-fpm: pool www"

dic 12 17:05:56 verty002-pc systemd[1]: Starting php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager...
dic 12 17:05:56 verty002-pc systemd[1]: Started php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager...
[lines 1-17/17 (END)...skipping...
● php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager
  Loaded: loaded (/usr/lib/systemd/system/php8.3-fpm.service; enabled; preset: enabled)
  Active: active (running) since Thu 2024-12-12 17:05:56 CET; 16min ago
    Docs: man:php-fpm8.3(8)
   Process: 1621 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run/php/php-fpm.sock /etc/php/8.3/fpm/pool.d/www.conf 83 (code=exited, status=0/SUCCESS)
  Main PID: 1594 (php-fpm8.3)
    Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
     Tasks: 3 (limit: 18401)
    Memory: 15.1M (peak: 16.1M)
      CPU: 119ms
     CGroup: /system.slice/php8.3-fpm.service
             ├─1594 "php-fpm: master process (/etc/php/8.3/fpm/php-fpm.conf)"
             ├─1619 "php-fpm: pool www"
             └─1620 "php-fpm: pool www"

dic 12 17:05:56 verty002-pc systemd[1]: Starting php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager...
dic 12 17:05:56 verty002-pc systemd[1]: Started php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager.
-
```

Nginx

Actualizamos repositorios

`sudo apt update`

Instalamos el nginx

`sudo apt install nginx -y`

Verificamos la versión que nos hemos descargado.

nginx-v

```
junle@Jun1nh0PC:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
```

Para poder utilizar php-fpm junto a Nginx tendremos que configurar un archivo que tendrá esta dirección:

/etc/nginx/sites-available/default

```
junle@Jun1nh0PC:~$ sudo nano /etc/nginx/sites-available/default
```

```
C/C++
#nginx config
server {
    listen 8080;
    server_name localhost;

    root /var/www/html; # Carpeta arrel dels teus fitxers
    index index.php index.html index.htm;

    # Bloc principal per a arxius estàtics i routing
    location / {
        try_files $uri $uri/ /server.php;
    }

    # Processar fitxers PHP
    location ~ \.php$ {

        fastcgi_split_path_info ^(/server.php)(/.+)$;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_index server.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }

    # Protegir arxius .htaccess
    location ~ /\.ht {
        deny all;
    }
}
```

```
}
```

Comandos importantes de Nginx

sudo systemctl enable nginx

```
ivan_rodriguez :: verty002-pc → /var/www/html » sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
ivan_rodriguez :: verty002-pc → /var/www/html »
```

sudo systemctl status nginx

```
ivan_rodriguez :: verty002-pc → ~ » sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Thu 2024-12-12 17:06:02 CET; 17min ago
    Docs: man:nginx(8)
 Process: 2366 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 2371 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 2374 (nginx)
   Tasks: 17 (limit: 18401)
  Memory: 12.9M (peak: 14.1M)
    CPU: 38ms
   CGroup: /system.slice/nginx.service
           ├─2374 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─2375 "nginx: worker process"
           ├─2376 "nginx: worker process"
           ├─2377 "nginx: worker process"
           ├─2378 "nginx: worker process"
           ├─2379 "nginx: worker process"
           ├─2381 "nginx: worker process"
           ├─2382 "nginx: worker process"
           ├─2383 "nginx: worker process"
           ├─2384 "nginx: worker process"
           ├─2385 "nginx: worker process"
           ├─2386 "nginx: worker process"
           ├─2387 "nginx: worker process"
           ├─2388 "nginx: worker process"
           ├─2389 "nginx: worker process"
           ├─2390 "nginx: worker process"
           └─2391 "nginx: worker process"

dic 12 17:06:02 verty002-pc systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
dic 12 17:06:02 verty002-pc systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

NOTA IMPORTANTE

```
ivan_rodriguez :: verty002-pc → ~ » cd /var/www/html/
ivan_rodriguez :: verty002-pc → /var/www/html » ls
database.php  database.php  index.nginx-debian.html  info.php  server.php  server.php.save  server_prueba.php
ivan_rodriguez :: verty002-pc → /var/www/html »
```

Si queremos utilizar php-fpm junto a nginx lo que tendremos que hacer es poner los archivos.php en un directorio en concreto, el cual si hemos seguido nuestra instalación tiene que ser **/var/www/html**

MariaDB

Descargamos los paquetes necesarios para trabajar con repositorios externo

```
sudo apt install software-properties-common dirmngr
```

Actualizamos los repositorios

```
sudo apt update
```

Mostramos información detallada de lo que hemos instalado y lo descargamos

```
apt-cache show mariadb-server
sudo apt install mariadb-server
```

Miramos si la versión que nos hemos descargado de MariaDB es la correcta, podemos usar cualquiera de los siguientes comandos

```
mariadb --version
```

```
mysql --version
```

```
junLe@JunInhoPC:~$ mysql --version
mysql  Ver 15.1 Distrib 10.11.8-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper
```

Configuración de MariaDB

```
DROP USER 'root'@'localhost';
```

```
CREATE USER 'root'@'localhost' IDENTIFIED BY 'telematik'; (donde pone telematik,es la contraseña que usaremos esta, para no tener que poner una personal )
```

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

2. Servidor

En esta nueva y última versión del proyecto, hemos decidido cambiar la estructura del servidor por completo. Escuchando las indicaciones del profesor y a criterio nuestro, creemos que la mejor manera de implementar el servidor es la que explicaremos a continuación.

```
Java
<?php

// Configuración de la conexión a la base de datos
header('Access-Control-Allow-Origin: *');      //es per poder accedir desde
// cualsevol client
header("Access-Control-Allow-Headers:           Origin,           X-Requested-With,
Content-Type, Accept");
header('Access-Control-Allow-Methods: GET, POST, PUT, DELETE');

// Establecer el encabezado de la respuesta
header('Content-Type: application/json');

// Configuración de la conexión a la base de datos
include("database.php");

// Extraer la tabla del path de la URL
$path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$pathParts = explode('/', trim($path, '/'));
$table = end($pathParts);
$uid=$pathParts[count($pathParts)-2];

// Validar el nombre de la tabla
$allowedTables = ['marks', 'tasks', 'timetables', 'students'];
if (!in_array(strtolower($table), $allowedTables)) {
    http_response_code(400);
    die(json_encode(['error' => 'Invalid table name']));
}

// Inicializar la consulta base
```

```

$query = "SELECT * FROM `{$table}`";
$whereConditions = [];

// Analizar los parámetros de la consulta
$queryString = parse_url($_SERVER['REQUEST_URI'], PHP_URL_QUERY);
$queryParams = [];
if ($queryString) {
    parse_str($queryString, $queryParams);
}

// Registro de depuración de los parámetros
error_log("Parsed Query Parameters: " . print_r($queryParams, true));

// Procesar filtros y parámetros de control
$limit = null; // Para almacenar el valor de limit, si existe

foreach ($queryParams as $field => $conditions) {
    if ($field === 'limit') {
        // Procesar el parámetro LIMIT
        $limit = intval($conditions); // Convertir a entero
        continue; // Saltar al siguiente parámetro
    }

    if (is_array($conditions)) {
        // Filtros con operadores (ej: date[gte], mark[lt])
        foreach ($conditions as $operator => $value) {
            if ($value === 'now') {
                $value = 'CURDATE()'; // Reemplazar 'now' con la función de
MySQL
            } else {
                $value = $mysqli->real_escape_string($value); // Escapar
valores para seguridad
                $value = "'{$value}'"; // Agregar comillas a valores literales
            }
        }
    }
}

switch ($operator) {

```

```

        case 'lt': $whereConditions[] = "`$field` < $value ";
break;
        case 'lte': $whereConditions[] = "`$field` <= $value";
break;
        case 'gt': $whereConditions[] = "`$field` > $value"; break;
        case 'gte': $whereConditions[] = "`$field` >= $value";
break;
        case 'eq': $whereConditions[] = "`$field` = $value"; break;
    }
}
} else {
    // Filtro directo (ej: date=2024-11-25)
    $value = $mysqli->real_escape_string($conditions);
    $whereConditions[] = "`$field` = '$value'";
}
}

if($table=='marks'){
    $whereConditions[] = " uid = '$uid'";
}

// Construir la cláusula WHERE si hay condiciones
if (!empty($whereConditions)) {
    $query .= " WHERE " . implode(' AND ', $whereConditions);

}

// Añadir cláusula LIMIT si está especificado
if ($limit !== null) {
    $query .= " LIMIT $limit";
}

// Registro de depuración de la consulta final
error_log("Final Generated SQL Query: $query");

// Ejecutar la consulta
$result = $mysqli->query($query);

```

```
if ($result) {
    // Obtener resultados
    $data = $result->fetch_all(MYSQLI_ASSOC);

    // Registro de depuración de los datos obtenidos
    error_log("Retrieved data: " . print_r($data, true));

    // Responder con JSON
    echo json_encode($data, JSON_PRETTY_PRINT);
} else {
    // Manejar errores de consulta
    http_response_code(500);
    error_log("Query error: " . $mysqli->error);
    echo json_encode([
        'error' => 'Database query failed',
        'details' => $mysqli->error
    ]);
}
// Cerrar la conexión a la base de datos
$mysqli->close();
?>
```

3. Cliente

Protocolo http GET

- Cliente solicita información específica desde la base de datos a través del servidor
- Sintaxis de la solicitud GET especifica que tipo de datos quiere a partir de parámetro URL.

```
GET /timetables?day=Fri&hour[gt]=08:00 HTTP/1.1
```

// timetables el horario
// day=Fri la fecha es el viernes
//hour[gt]=08:00 a partir de las 8
//HTTP/1.1 es la version del http

Host: servidor.com

// nombre del servidor en el host

Accept: application/json

//accept lo que hace es indicar que espera una respuesta de tipo Jherson (JSON)

Alumno: (student_id lo podemos encontrar como uid)

- El alumno se identifica con student_id, la student_id se envía al servidor y lo busca en una tabla interna que contiene (nombre_estudiante, student_id)
 - Si el student_id **SI** existe, entonces se printea por el LCD “Welcome NOMBRE_ESTUDIANTE”
 - Si el student_id **NO** existe, entonces debe mostrar un mensaje de error
- Dentro del query hay **tres** tipos de tablas:
 - **tasks:** fecha de entrega / asignatura / nombre de la entrega.

task_id	subject	name	date
1	AST	Práctica 1	2018/09/20
2	PBE	Project Plan	2018/09/20
3	AST	TD	2018/09/20
4	PBE	Requirement Specifications	2018/09/22
5	PBE	Critical Design Report	2018/10/25
6	AST	TD	2018/10/25
7	PBE	Final Report	2018/10/25
8	AST	Práctica 4	2018/10/25

- **timetables**: dia que tienes clase (lunes, martes...) / hora de la clase (la que comienza) / asignatura / clase en la que la haces.

day	hour	subject	room
Tue	08:00:00	TD	A4-105
Tue	10:00:00	PBE	A4-105
Tue	11:00:00	DSBM	A4-105
Tue	12:00:00	RP	A4-105
Wed	08:00:00	TD	A4-105
Wed	10:00:00	PBE	A4-105
Thu	08:00:00	PBE	A4-105
Thu	10:00:00	TD	A4-105
Fri	08:00:00	RP	A4-105
Fri	10:00:00	TD	A4-105
Fri	11:00:00	DSBM	A4-105
Mon	08:00:00	TD	A4-105
Mon	10:00:00	PBE	A4-105
Mon	12:00:00	Lab DSBM	C5-005

- **marks**: asignatura / nombre del trabajo o si es una examen (parcial 1, lab 2, puzzle 3...) / nota que tienes

subject	name	mark
AST	control teoria	7.5
AST	Lab1	7.5
AST	Lab2	4.7
AST	final	8.8
PBE	puzzle1	8.5
PBE	puzzle2	8.4
PBE	control	7.3
PBE	CDR	8
PBE	FR	9

En el cliente Python, puedes usar la biblioteca `requests` para hacer la solicitud al servidor PHP y recibir la respuesta. Luego, la respuesta JSON puede ser convertida en un formato que el cliente pueda usar en la interfaz gráfica.

Código de cliente en la RPI

```
Python
import threading
import gi
import json
import requests
import time
from primer_puzzle_adafruit import RfidReader
#from i2clcd import i2clcd
gi.require_version("Gtk", "3.0")
from gi.repository import Gtk, GLib, Pango, Gdk
```

```

class CourseManager(Gtk.Window):
    def __init__(self):
        super().__init__(title="Course Manager")
        self.server = "10.42.0.1"
        self.port = "8080"
        self.timer = None
        self.inactivity_timeout = 50 # 50 seconds inactivity timeout
        #self.lcd = i2clcd(1,0x27,20)
        #self.lcd.init()
        self.rfid_reader = RfidReader()
        self.uid = None

        #apply css to app
        self.apply_css()

        # Window configuration
        self.set_border_width(10)
        self.set_default_size(700, 600)
        self.connect("destroy", self.on_destroy)

        # Main container: using Gtk.Stack to switch between screens
        self.stack = Gtk.Stack()
        self.add(self.stack)

        # Login screen

        self.label = Gtk.Label() #aixo es pot treure crec
        self.label.set_size_request(100,100) #aixo es pot treure crec
        self.login_box = Gtk.Box(orientation=Gtk.Orientation.VERTICAL,
                               spacing=10)
        self.login_box.get_style_context().add_class("login-box") # Add
        custom CSS class
        self.stack.add_named(self.login_box, "login")

        self.login_label = Gtk.Label(label="PLEASE, LOGIN WITH YOUR
UNIVERSITY CARD")

```

```

        self.login_label.get_style_context().add_class("login-label")
        self.login_box.pack_start(self.login_label, True, True, 0)

        # Query screen
        self.query_box = Gtk.Box(orientation=Gtk.Orientation.VERTICAL,
spacing=10)
        self.query_box.get_style_context().add_class("query-box")    # Add
custom CSS class
        self.stack.add_named(self.query_box, "query")

        self.welcome_label = Gtk.Label(label="Welcome")
        self.welcome_label.get_style_context().add_class("welcome-label")
        self.query_box.pack_start(self.welcome_label, False, False, 0)

        self.query_entry = Gtk.Entry()
        self.query_entry.set_placeholder_text("Enter your query")
        self.query_entry.get_style_context().add_class("query-entry")
        self.query_entry.connect("activate", self.on_query)
        self.query_box.pack_start(self.query_entry, False, False, 0)

        self.logout_button = Gtk.Button(label="Logout")
        self.logout_button.get_style_context().add_class("logout-button")
        self.logout_button.connect("clicked", self.on_logout)
        self.query_box.pack_start(self.logout_button, False, False, 0)

        # Initially show login screen
        self.stack.set_visible_child_name("login")

        # Start RFID reader thread
        self.reader_thread = threading.Thread(target=self.read_uid_thread,
daemon=True)
        self.reader_thread.start()

        self.show_all()

def apply_css(self):
    # Load CSS

```

```

css_provider = Gtk.CssProvider()
css_provider.load_from_path("styles.css") # Ruta al archivo CSS

# Apply CSS to the default screen
screen = Gdk.Screen.get_default()
style_context = Gtk.StyleContext()
style_context.add_provider_for_screen(
    screen,
    css_provider,
    Gtk.STYLE_PROVIDER_PRIORITY_APPLICATION
)

def read_uid_thread(self):
    while True:
        uid = self.rfid_reader.read_uid()
        if uid:
            GLib.idle_add(self.authenticate_user, uid)

        time.sleep(0.1)

def authenticate_user(self, uid):
    # Construct the URL to query students table with the specific UID
    self.uid = uid
    url = f"http://{{self.server}}:{{self.port}}/server.php/students?uid={{uid}}"
    try:
        response = requests.get(url, timeout=5)
        response.raise_for_status()
        user_data = response.json()

        if user_data and isinstance(user_data, list) and len(user_data) > 0:
            # Assuming the first result contains the name
            name = user_data[0].get('name', 'Unknown')
            self.current_name = name # Store the name for later use
            self.update_welcome_screen(name)
    except:
        pass

```

```

        self.update_login_label("User not found !!TRY AGAIN", "red")
    except requests.exceptions.RequestException as e:
        self.update_login_label(f"Connection error: {str(e)}", "red")
    except ValueError:
        self.update_login_label("Invalid server response", "red")

    def update_welcome_screen(self, name, is_error=False):
        # Default welcome message with name
        welcome_message = f"Hello {name} !!! It's always nice to see you
again :)"

        if is_error:
            # If it's an error, show the error message in red
            self.welcome_label.set_markup(f'<span foreground="red">Incorrect
query. Please try again.</span>')
        else:
            # Default text color for welcome message
            self.welcome_label.set_text(welcome_message)

        self.stack.set_visible_child_name("query")
        self.start_inactivity_timer()

    def update_login_label(self, text, color="red"):
        self.login_label.set_markup(f'<span
foreground="{color}">{text}</span>')

    def on_query(self, widget):
        query_text = self.query_entry.get_text().strip()
        if query_text:
            url      =
f"http://self.server}:{self.port}/server.php/{self.uid}/{query_text}"
            threading.Thread(target=self.perform_query, args=(url,)).start()
        else:
            self.update_welcome_label("Query is empty!", "red")

    def perform_query(self, url):

```

```

try:
    response = requests.get(url, timeout=5)
    response.raise_for_status()
    result = response.json()
    GLib.idle_add(self.on_response, result)
except requests.exceptions.RequestException as e:
    GLib.idle_add(self.on_response, None) # Pass None to trigger
error handling

def on_response(self, result):
    # Remove previous treeview if it exists
    if hasattr(self, "treeview"):
        self.treeview.destroy()

    # Check if result is None or empty/invalid
    if not result or not isinstance(result, list):
        # Update the welcome label with an error message
        self.update_welcome_screen(self.current_name, is_error=True)
        return

    # If result is valid, create the table and restore welcome message
    keys = list(result[0].keys())
    if not keys:
        self.update_welcome_screen(self.current_name, is_error=True)
        return

    # Restore welcome message
    self.update_welcome_screen(self.current_name)

    # Create table as before
    self.liststore = Gtk.ListStore(*[str] * len(keys))

    for item in result:
        row = [str(item.get(key, "")) for key in keys]
        self.liststore.append(row)

    self.treeview = Gtk.TreeView(model=self.liststore)

```

```

        for i, column_title in enumerate(keys):
            renderer = Gtk.CellRendererText()
            column = Gtk.TreeViewColumn(column_title, renderer, text=i)
            column.set_resizable(True)
            column.set_alignment(0.5)
            self.treeview.append_column(column)

        self.treeview.modify_font(Pango.FontDescription("Arial 12"))
        self.treeview.set_rules_hint(True)

        self.query_box.pack_start(self.treeview, True, True, 0)
        self.query_box.show_all()

    def on_logout(self, button):

        self.stop_timer()
        # self.lcd.clear()
        # Reset the UID
        self.uid = None
        # Reset the current name
        self.current_name = None
        # Remove the treeview if it exists
        if hasattr(self, "treeview"):
            self.treeview.destroy()
            del self.treeview
        # Remove the liststore if it exists
        if hasattr(self, "liststore"):
            del self.liststore
        # Reset query
        self.query_entry.set_text("")
        # Switch back to login screen
        self.stack.set_visible_child_name("login")
        self.login_label.set_text("PLEASE, LOGIN WITH YOUR UNIVERSITY CARD")
        self.welcome_label.set_text("")

```

```
def start_inactivity_timer(self):
    self.stop_timer()
    self.timer = GLib.timeout_add_seconds(self.inactivity_timeout,
self.on_timeout)

def on_timeout(self):
    self.on_logout(None)
    self.stop_timer()

def stop_timer(self):
    if self.timer:
        GLib.source_remove(self.timer)
    self.timer = None

def on_destroy(self, widget):
    Gtk.main_quit()

def main():
    app = CourseManager()
    Gtk.main()

if __name__ == "__main__":
    main()
```

4. Database

El código del database se ha mantenido constante a lo largo de las diferentes versiones del proyecto.

Para importar nuestro archivo sql, en mariaDB debemos utilizar el siguiente comando:

```
ivan_rodriguez :: vertx002-pc ~ » sudo mysql -u root -p atenea_pbe < /home/ivan_rodriguez/atenea_pbe.sql
```

Indicando el nombre de la base de datos y la ubicación del archivo sql.

```
Java
-----
--DB AND SERVER INFORMATION-----
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;

-----
-- TABLE CREATION --
--(50) is the max num of characters that username can contain this is an
arbitrary number
CREATE TABLE students (
    name VARCHAR(50) NOT NULL, -- name of the student
    uid VARCHAR(20) NOT NULL -- this is going to be the uid, each student
has an associated uid
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE tasks (
    date DATE NOT NULL, -- deadline
    subject VARCHAR(50) NOT NULL, -- the length of the string are not going
to be less than 50 to be safe
    name VARCHAR(50) NOT NULL -- name of the work
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE timetable (
```

```

    day ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday') NOT
NULL,
    hour TIME NOT NULL,
    subject VARCHAR(50) NOT NULL,
    class VARCHAR(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE marks (
    uid VARCHAR(10) NOT NULL,
    subject VARCHAR(50) NOT NULL,
    name VARCHAR(100) NOT NULL, -- name of the exam or task
    mark DECIMAL(4,2) NOT NULL -- 4 digits in total, 2 reserved for decimals
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

----- INSERT INFORMATION-----

```

INSERT INTO students (name, uid) VALUES
('Ivan Rodriguez', '4BD56667'),
('Junle Wang', '5F63B81E'),
('Blanca Calero', 'B3C4BA24'),
('Mariona Capdevila', '72B6E200'),
('Roger Garcia', '24BAC4B3');

```

```

INSERT INTO tasks (date, subject, name) VALUES
('2024-11-25', 'DSBM', 'Practica 5'),
('2024-10-21', 'DSBM', 'Practica 2'),
('2024-09-27', 'RP', 'Control 1'),
('2024-10-13', 'RP', 'Control 2'),
('2024-11-17', 'RP', 'Control 3'),
('2024-12-30', 'PSAVC', 'Practica 1'),
('2024-12-11', 'PSAVC', 'Pràctica 2'),
('2024-12-27', 'PBE', 'CDR'),
('2024-11-13', 'PBE', 'Entrega PUZZLE2');

```

```

INSERT INTO timetables (day, hour, subject, class)VALUES
('Monday', '08:00:00', 'RP_LAB', 'Aula D3006'),
('Monday', '10:00:00', 'RP_TEORIA', 'Aula A4105'),

```

```

('Monday', '11:00:00', 'DSBM_TEORIA', 'Aula A4105'),
('Tuesday', '08:00:00', 'PSAVC_TEORIA', 'Aula A4105'),
('Tuesday', '11:00:00', 'TD_TEORIA', 'Aula A4105'),
('Wednesday', '08:00:00', 'PBE_LAB', 'Aula A4105'),
('Thursday', '08:00:00', 'PBE_TEORIA', 'Aula A4105'),
('Thursday', '10:00:00', 'RP_TEORIA', 'Aula A4105'),
('Thursday', '12:00:00', 'DSBM_LAB', 'Aula C5S101A'),
('Friday', '08:00:00', 'DSBM_TEORIA', 'Aula A4105'),
('Friday', '10:00:00', 'PSAVC', 'Aula A4105'),
('Friday', '12:00:00', 'TD', 'Aula A4105');

```

```

INSERT INTO marks (uid, subject, name, mark)VALUES
('4BD56667', 'DSBM', 'Examen Parcial', 3.75),
('4BD56667', 'RP', 'Examen Parcial', 2),
('4BD56667', 'PSAVC', 'Examen Parcial', 2),
('4BD56667', 'TD', 'Examen Parcial', 3),
('4BD56667', 'PBE', 'Examen Parcial', 8),
('5F63B81E', 'DSBM', 'Examen Parcial', 6),
('5F63B81E', 'RP', 'Examen Parcial', 0.7),
('5F63B81E', 'PSAVC', 'Examen Parcial', 2.8),
('5F63B81E', 'TD', 'Examen Parcial', 4.75),
('5F63B81E', 'PBE', 'Examen Parcial', 8.7),
('B3C4BA24', 'DSBM', 'Examen Parcial', 5.7),
('B3C4BA24', 'RP', 'Examen Parcial', 1.2),
('B3C4BA24', 'PSAVC', 'Examen Parcial', 6.6),
('B3C4BA24', 'TD', 'Examen Parcial', 3.4),
('B3C4BA24', 'PBE', 'Examen Parcial', 7.2),
('72B6E200', 'DSBM', 'Examen Parcial', 2.5),
('72B6E200', 'RP', 'Examen Parcial', 3.75),
('72B6E200', 'PSAVC', 'Examen Parcial', 8.5),
('72B6E200', 'TD', 'Examen Parcial', 4),
('72B6E200', 'PBE', 'Examen Parcial', 5.6),
('24BAC4B3', 'DSBM', 'Examen Parcial', 7),
('24BAC4B3', 'RP', 'Examen Parcial', 3),
('24BAC4B3', 'PSAVC', 'Examen Parcial', 4.5),
('24BAC4B3', 'TD', 'Examen Parcial', 2),
('24BAC4B3', 'PBE', 'Examen Parcial', 6);

```

5. Conexiones

5.1. Conexión cliente - servidor

Configuración de nginx para que admita archivos php y sea capaz de parsear la url de la forma que nosotros queremos

Para consultar y editar este archivo tenemos que usar el siguiente comando:

```
ivan_rodriguez :: verty002-pc ~ » sudo nano /etc/nginx/sites-available/default

#nginx config

server {
    listen 8080;
    server_name localhost;

    root /var/www/html; # Carpeta arrel dels teus fitxers
    index index.php index.html index.htm;

    # Bloc principal per a arxius estàtics i routing
    location / {
        try_files $uri $uri/ /server.php;
    }

    # Processar fitxers PHP
    location ~ \.php$ {

        fastcgi_split_path_info ^(/server.php)(/.+)$;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_index server.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
}
```

```
}

# Protegir arxius .htaccess
location ~ /\.ht {
    deny all;
}
```

5.2. Conexión servidor - database

Para hacer la conexión de la base de datos con el servidor, hemos implementado este código en un fichero a parte:

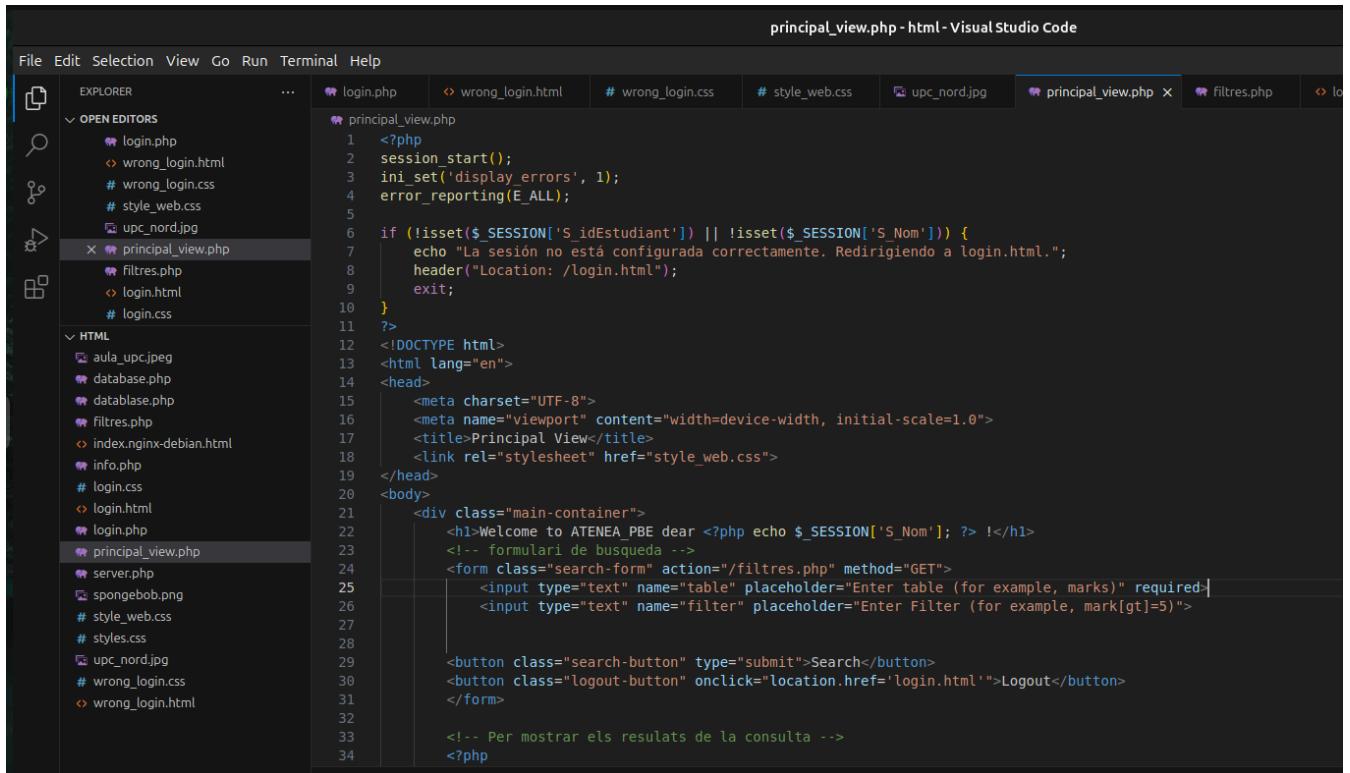
```
C/C++
<?php

$host = 'localhost';
$username = 'root';
$password = 'telematik';
$database = 'atenea_pbe';

// Establecer la conexión a la base de datos
$mysqli = new mysqli($host, $username, $password, $database);
if ($mysqli->connect_error) {
    die(json_encode([
        'error' => 'Database connection failed',
        'details' => $mysqli->connect_error
    ]));
}
?>
```

6. Cliente web

Puesto que para hacer el cliente necesitamos controlar más archivos la forma más cómoda de trabajar es con una IDE como VS Code:



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows a tree view of files and folders. The 'principal_view.php' file is selected in the tree.
- Code Editor:** Displays the PHP code for 'principal_view.php'. The code includes session handling, error reporting, and HTML structure for a search form.

```
<?php
session_start();
ini_set('display_errors', 1);
error_reporting(E_ALL);

if (!isset($_SESSION['S_idEstudiant']) || !isset($_SESSION['S_Nom'])) {
    echo "La sesión no está configurada correctamente. Redirigiendo a login.html.";
    header("Location: /login.html");
    exit;
}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Principal View</title>
    <link rel="stylesheet" href="style_web.css">
</head>
<body>
    <div class="main-container">
        <h1>Welcome to ATENEA_PBE dear <?php echo $_SESSION['S_Nom']; ?> !</h1>
        <!-- formulario de búsqueda -->
        <form class="search-form" action="/filtres.php" method="GET">
            <input type="text" name="table" placeholder="Enter table (for example, marks)" required=>
            <input type="text" name="filter" placeholder="Enter Filter (for example, mark[gt]=5)">
            <button class="search-button" type="submit">Search</button>
            <button class="logout-button" onclick="location.href='login.html'">Logout</button>
        </form>
        <!-- Per mostrar els resultats de la consulta -->
    </div>
</body>
<?php
```

Para hacer esto lo que debemos hacer es dar permiso al directorio /var/www/html (aquí es donde tenemos todos los archivos) y ya podemos trabajar y guardar el código sin tener que identificarnos cada vez que queremos guardar algo .

```
WELCOME TO UBUNTU 24.04 TERMINAL
ivan_rodriguez :: verty002-pc ~ » sudo chown -R ivan_rodriguez /var/www/html
[sudo] contraseña para ivan_rodriguez:
ivan_rodriguez :: verty002-pc ~ »
```

7. login.php: código para iniciar la sesión del nuevo usuario. En caso de que sea correcto se mostrará la siguiente pantalla que corresponde a principal_view.php. Si las credenciales son incorrectas, se mostrará por pantalla un mensaje de “Wrong Credentials” a través del archivo wrong_login.html. Si se desea cerrar la sesión, se toma el botón de logout y se dirige la web al documento inicial de login.html

C/C++

```

<?php
session_start();
ini_set('display_errors', 1);
error_reporting(E_ALL);

include("database.php");

if (!empty($_POST['username']) && !empty($_POST['password'])) {
    $username = $mysqli->real_escape_string($_POST['username']);
    $password = $mysqli->real_escape_string($_POST['password']);

    $sql = "SELECT * FROM students WHERE name=' $username ' AND
uid=' $password '";
    $result = $mysqli->query($sql);

    if ($result && $row = $result->fetch_assoc()) {
        // Guardar credenciales
        $_SESSION['S_idEstudiant'] = $row['uid'];
        $_SESSION['S_Nom'] = $row['name'];

        // Redirigir
        header("Location:/principal_view.php");
        exit;
    } else {
        // Redirigir a página de error
        header("Location: /wrong_login.html");
        exit;
    }
} else {
    // Redirigir al formulario de login
    header("Location: /login.html");
    exit;
}
?>

```

8. **principal_view.php**: Se tendrá la opción de escribir una query para realizar una búsqueda o un botón de logout para cerrar la sesión. De cara a la parte gráfica hemos implantado dos buscadores, uno para las tablas y otro para los filtros, de esta forma es más fácil escribir la consulta, ya que nos ahorraremos el interrogante.

```
C/C++
<?php
session_start();
ini_set('display_errors', 1);
error_reporting(E_ALL);

// Verificar si la sesión contiene los datos esperados
if (!isset($_SESSION['S_idEstudiant']) || !isset($_SESSION['S_Nom'])) {
    echo "La sesión no está configurada correctamente.
Redirigiendo a login.html.";
    header("Location: /login.html");
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Principal View</title>
    <link rel="stylesheet" href="style_web.css">
</head>
<body>
    <div class="main-container">
        <h1>Welcome to ATNEA_PBE dear <?php echo
$_SESSION['S_Nom']; ?> !</h1>
```

```

<!-- Pritear error de tabla -->
<?php
    if (isset($_SESSION['errorTable']) &&
$_SESSION['errorTable'] === true) {
        echo "<p style='color: red;'>Table Error, try again
please.</p>";
        // Limpiar el valor del booleano para evitar
mostrarlo de nuevo
        unset($_SESSION['errorTable']);
    }
?>

<!-- Formulario de búsqueda -->
<form class="search-form" action="/filtres.php"
method="GET">
    <input type="text" name="table" placeholder="Enter
table (for example, marks)" required>
    <input type="text" name="filter" placeholder="Enter
Filter (for example, mark[gt]=5)">
    <button class="search-button"
type="submit">Search</button>
    <button class="logout-button"
onclick="location.href='login.html'">Logout</button>
</form>

<!-- Mostrar resultados -->
<?php
if (isset($_SESSION['S_result'])) {
    $resultados = json_decode($_SESSION['S_result'],
true);

```

```

if (!empty($resultados)) {
    echo "<table border='1'>";
    echo "<thead>";
    foreach (array_keys($resultados[0]) as $columna)
    {
        echo "<th>$columna</th>";
    }
    echo "</thead>";
    echo "<tbody>";
    foreach ($resultados as $fila) {
        echo "<tr>";
        foreach ($fila as $valor) {
            echo "<td>$valor</td>";
        }
        echo "</tr>";
    }
    echo "</tbody>";
    echo "</table>";
} else {
    echo "<p>No results found.</p>";
}

// Limpiar la sesión para evitar resultados
duplicados
unset($_SESSION['S_result']);
}

?>
</div>
</body>
</html>

```

9. **filtres.php:** Se llama al apretar el botón de send, hará la búsqueda en la base de datos y devolverá a la página principal el resultado en json donde se decodificará y se mostrará con una tabla por pantalla. Es básicamente el código del servidor.php adaptado a la página web.

C/C++

```
<?php

session_start();
ini_set('display_errors', 1);
error_reporting(E_ALL);

header('Access-Control-Allow-Origin: *');
header("Access-Control-Allow-Headers: Origin, X-Requested-With,
Content-Type, Accept");
header('Access-Control-Allow-Methods: GET, POST, PUT, DELETE');
header('Content-Type: application/json');

include("database.php");

// Extraer la tabla del parámetro GET
$table = isset($_GET['table']) ? $_GET['table'] : null;
$filterString = isset($_GET['filter']) ? $_GET['filter'] : null;
$uid = $_SESSION['S_idEstudiant'];

// Validar el nombre de la tabla
$allowedTables = ['marks', 'tasks', 'timetables', 'students'];
if (!$table || !in_array(strtolower($table), $allowedTables)) {
    http_response_code(400);
    $_SESSION['errorTable'] = true;
```

```

echo "ERROR: Tabla no válida";
header("Location: principal_view.php");
exit;
}

// Inicializar la consulta base
$query = "SELECT * FROM `{$table}`";
$whereConditions = [];

// Procesar filtros
$queryParams = [];
if ($filterString) {
    parse_str($filterString, $queryParams);
    $_SESSION['errortabla'] = false;
    foreach ($queryParams as $field => $conditions) {
        if ($field === 'limit') {
            $limit = intval($conditions); // Procesar el
            parámetro LIMIT
            continue;
        }

        if (is_array($conditions)) {
            foreach ($conditions as $operator => $value) {
                $value = $mysqli->real_escape_string($value);
                $value = "'{$value}'";
                switch ($operator) {
                    case 'lt': $whereConditions[] = "{$field} < $value"; break;
                    case 'lte': $whereConditions[] = "{$field} <= $value"; break;
                }
            }
        }
    }
}

```

```

        case 'gt': $whereConditions[] = "`$field` >
$value"; break;
        case 'gte': $whereConditions[] = "`$field` >=
$value"; break;
        case 'eq': $whereConditions[] = "`$field` =
$value"; break;
    }
}
} else {
    $value = $mysqli->real_escape_string($conditions);
    $whereConditions[] = "`$field` = '$value'";
}
}

if ($table === 'marks') {
    $whereConditions[] = "`uid` = '$uid'";
}

// Construir la cláusula WHERE si hay condiciones
if (!empty($whereConditions)) {
    $query .= " WHERE " . implode(' AND ', $whereConditions);
}

// Añadir cláusula LIMIT si está especificado
if (isset($limit)) {
    $query .= " LIMIT $limit";
}

// Ejecutar la consulta
$result = $mysqli->query($query);

```

```

if ($result) {
    $data = $result->fetch_all(MYSQLI_ASSOC);

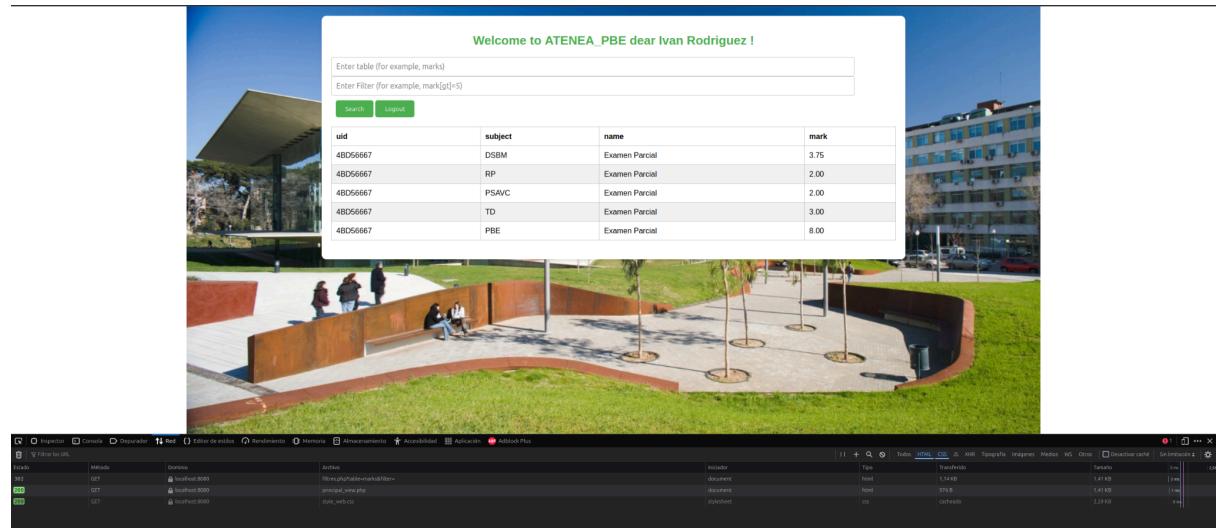
    // Guardar resultados en la sesión y redirigir de vuelta a
    principal_view.php

    $_SESSION['S_result'] = json_encode($data);
    header('Location: /principal_view.php');
    exit;
} else {
    http_response_code(500);
    echo json_encode([
        'error' => 'Database query failed',
        'details' => $mysqli->error
    ]);
}

$mysqli->close();
?>

```

Para comprobar que todo funcionaba a medida que íbamos implementando el código y para ver si las redirige eran correctas, hemos utilizado la herramienta de la tecla F12 en el apartado de red



De esta forma tenemos información sobre el estado y el método entre otras cosas.

```
▶ GET http://localhost:8080/principal_view.php

Estado          200 OK (...)
Versión        HTTP/1.1
Transferido    976 B (tamaño 1,41 KB)
Política de referencia strict-origin-when-cross-origin
Prioridad de la solicitud Highest
Resolución DNS Sistema
```

10. Ficheros CSS

- **style_web.css**

C/C++

```
/* style_web.css */

body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    margin: 0;
    padding: 0;
}

.main-container {
    max-width: 1200px;
    margin: 20px auto;
    padding: 20px;
    background: #ffffff;
    border-radius: 10px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    position: relative;
}
```

```
.main-container h1 {  
    color: #4caf50;  
    text-align: center;  
    margin-bottom: 20px;  
}  
  
.main-container button {  
    background: #4caf50;  
    border: none;  
    color: white;  
    padding: 10px 20px;  
    border-radius: 5px;  
    cursor: pointer;  
}  
  
.main-container button:hover {  
    background: #45a049;  
}  
  
.main-container table {  
    width: 100%;  
    border-collapse: collapse;  
    margin: 20px 0;  
    text-align: left;  
}  
  
.main-container table th,  
.main-container table td {  
    padding: 10px;  
    border: 1px solid #ccc;  
}  
  
.main-container table tr:nth-child(even) {  
    background: #f2f2f2;  
}  
  
.main-container form {
```

```

        margin-bottom: 20px;
    }

.main-container input {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
}

.main-container input[type="text"] {
    width: calc(100% - 110px);
    display: inline-block;
}

.main-container button[type="submit"] {
    margin-left: 10px;
    padding: 10px 20px;
}

.main-container h1 {
    font-size: 26px; /*es pot canviar la mida */
    color: #4caf50;
    text-align: center;
    margin-bottom: 20px;
}

.search-form button[type="submit"] {
    margin-top: 10px;
    padding: 10px 20px;
    background: #4caf50;
    border: None;
    color: white;
    border-radius: 5px;
    cursor: pointer;
}

```

```
.search-form button[type="submit"]:hover {  
    background: #45a049;  
}  
  
.search-form button[type="logout"] {  
    margin-top: 10px;  
    padding: 10px 20px;  
    background: #4caf50;  
    border: none;  
    color: white;  
    border-radius: 5px;  
    cursor: pointer;  
}  
  
.search-form button[type="logout"]:hover {  
    background: #45a049;  
}  
  
body {  
    background: url('upc_nord.jpg') no-repeat center center fixed;  
    background-size: contain;  
}
```

- login.css

C/C++

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f2f2f2;  
    margin: 0;  
    padding: 0;  
}  
  
.login-container {  
    max-width: 400px;  
    margin: 100px auto;  
    padding: 20px;  
    background: #ffffff;  
    border-radius: 10px;  
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
    text-align: center;  
}  
  
.login-container h1 {  
    margin-bottom: 20px;  
    color: #4caf50;  
}  
  
.login-container input {  
    width: 90%;  
    padding: 10px;  
    margin: 10px 0;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    font-size: 16px;  
}  
  
.login-container button {
```

```

width: 95%;
padding: 10px;
background: #4caf50;
border: none;
color: white;
border-radius: 5px;
font-size: 16px;
cursor: pointer;
}

.login-container button:hover {
background: #45a049;
}

body {
background: url('aula_upc.jpeg') no-repeat center center fixed;
background-size: contain;
}

```

- wrong_login.css

```

C/C++
body {
font-family: Arial, sans-serif;
background-color: #f2f2f2;
margin: 0;
padding: 0;
display: flex;
justify-content: center;
align-items: center;
height: 20vh;
}

h2 {

```

```

max-width: 400px;
margin: 10px auto;
padding: 20px;
background: #ffffff;
border-radius: 10px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
text-align: center;
color: #ff0000;
}

a {
display: block;
width: 95%;
max-width: 400px;
margin: 20px auto;
padding: 10px;
background: #4caf50;
border: none;
color: white;
border-radius: 5px;
font-size: 16px;
text-align: center;
text-decoration: none;
cursor: pointer;
}

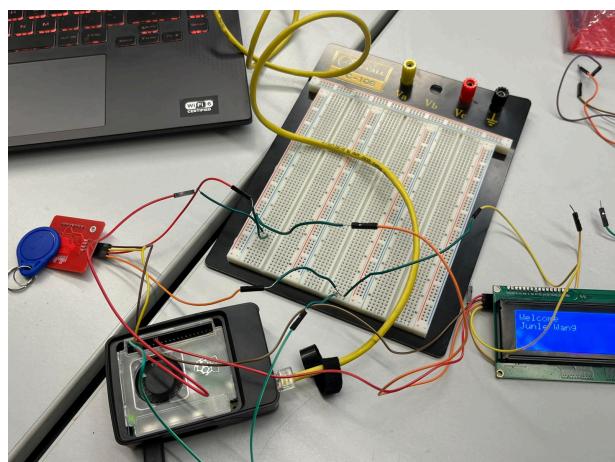
a:hover {
background: #45a049;
}

body {
background: url('aula_upc.jpeg') no-repeat center center fixed;
background-size: contain;
}

```

11. Resultado final

Cliente RPI

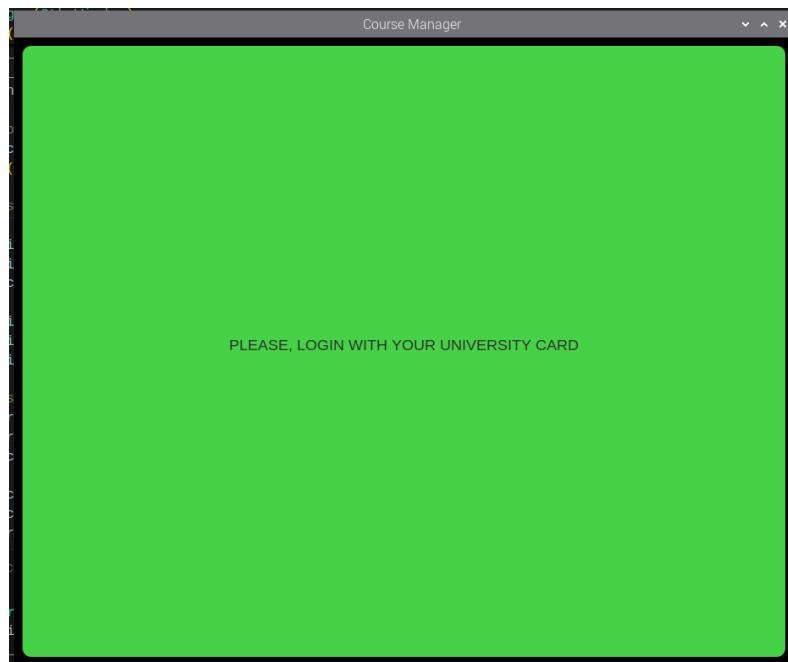


Este es un ejemplo de implementación conectando un lector NFC (PN532) y el LCD en la Raspberry, cabe decir que hemos tenido que usar una protoboard para poder conectarlos. Los dos periféricos están conectados a la RPI desde los puertos I2C es decir han tenido que compartir los puertos de SDA y SCL, lo único que hemos tenido que hacer es usar la protoboard con cables macho-macho, porque para conectar los periféricos a la RPI lo que usamos son cables hembra-hembra.

El funcionamiento del 'sistema' que hemos montado funciona a pesar de que comparten pines, ya que, el LCD está configurado con el puerto 0x27, mientras que el lector NFC en el puerto 0x24.

CDR

Ventana de login

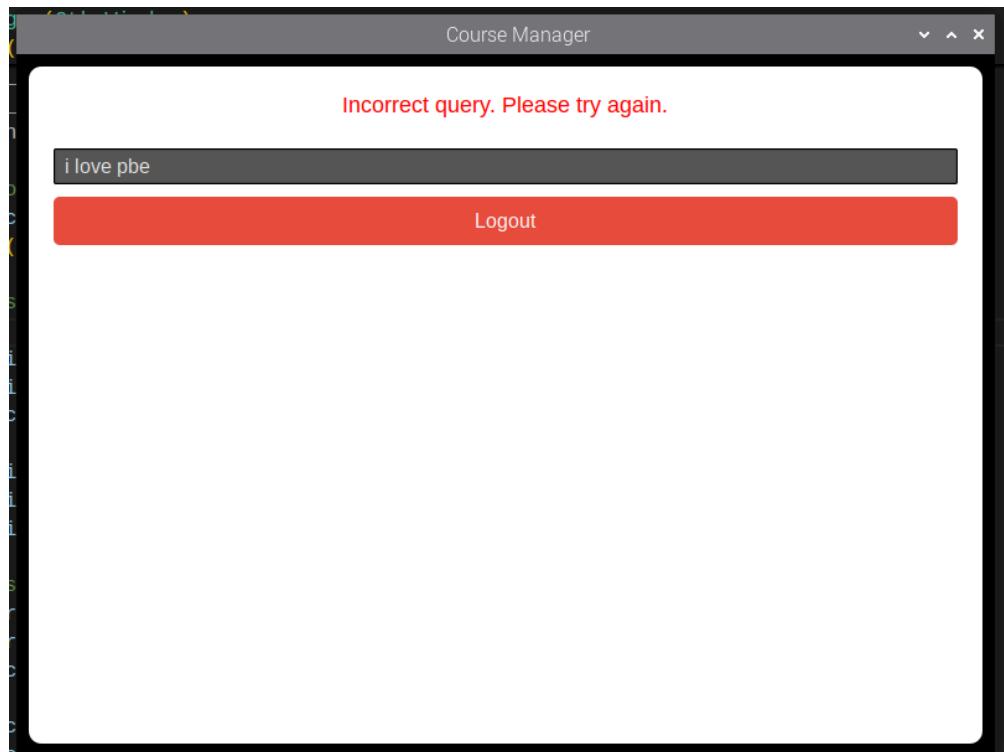


Ventana de consulta

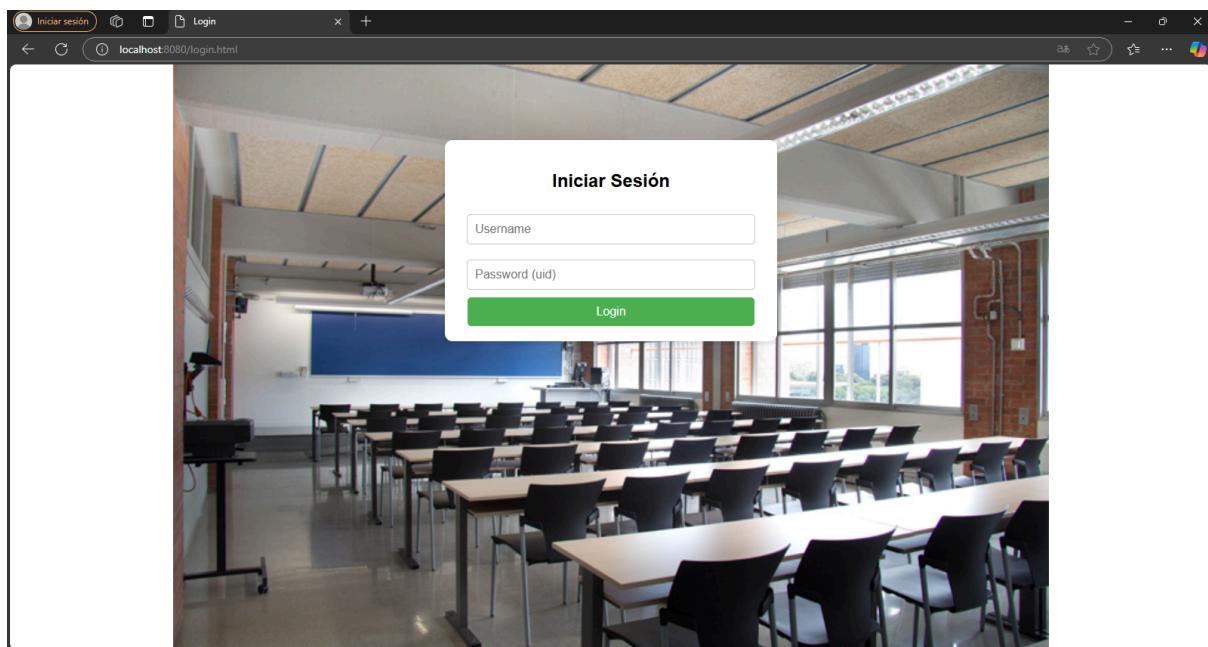
A screenshot of a computer window titled "Course Manager". The window displays a message "Hello Ivan Rodriguez !!! It's always nice to see you again :)" and a "timetables" section. The "timetables" section includes a "Logout" button and a table showing a weekly schedule. The table has columns for "day", "hour", "subject", and "class".

day	hour	subject	class
Monday	08:00:00	RP_LAB	Aula D3006
Monday	10:00:00	RP_TEORIA	Aula A4105
Monday	11:00:00	DSBM_TEORIA	Aula A4105
Tuesday	08:00:00	PSAVC_TEORIA	Aula A4105
Tuesday	11:00:00	TD_TEORIA	Aula A4105
Wednesday	08:00:00	PBE_LAB	Aula A4105
Thursday	08:00:00	PBE_TEORIA	Aula A4105
Thursday	10:00:00	RP_TEORIA	Aula A4105
Thursday	12:00:00	DSBM_LAB	Aula C5S101A
Friday	08:00:00	DSBM_TEORIA	Aula A4105
Friday	10:00:00	PSAVC	Aula A4105
Friday	12:00:00	TD	Aula A4105

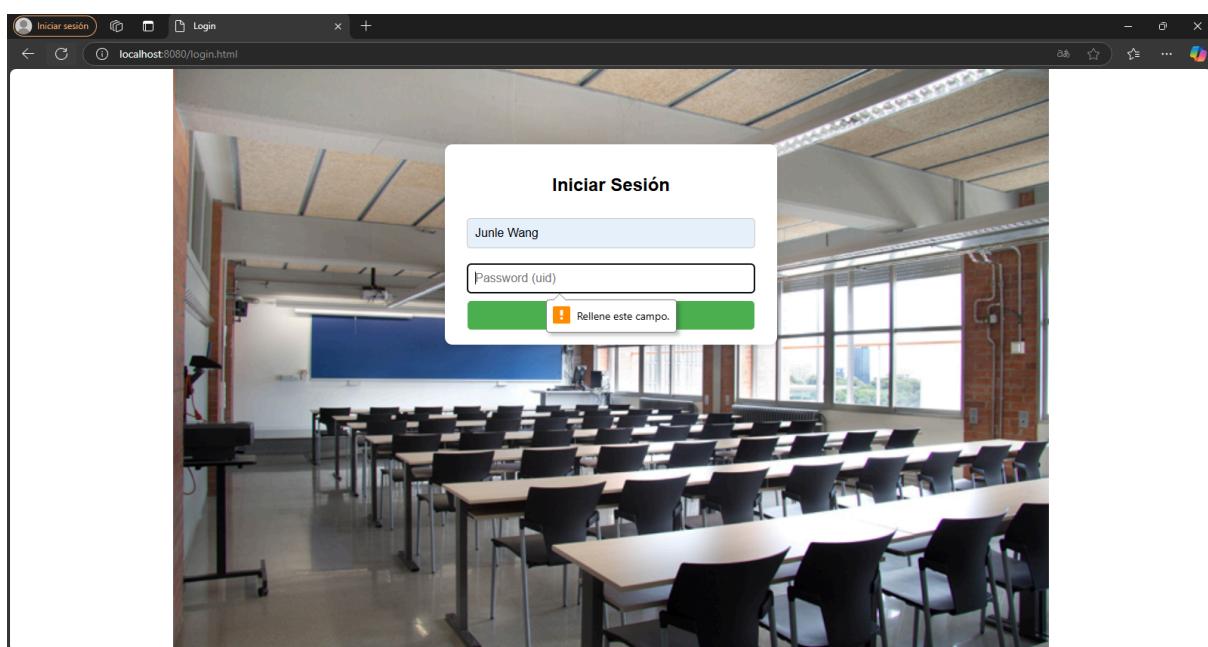
Mensaje de error



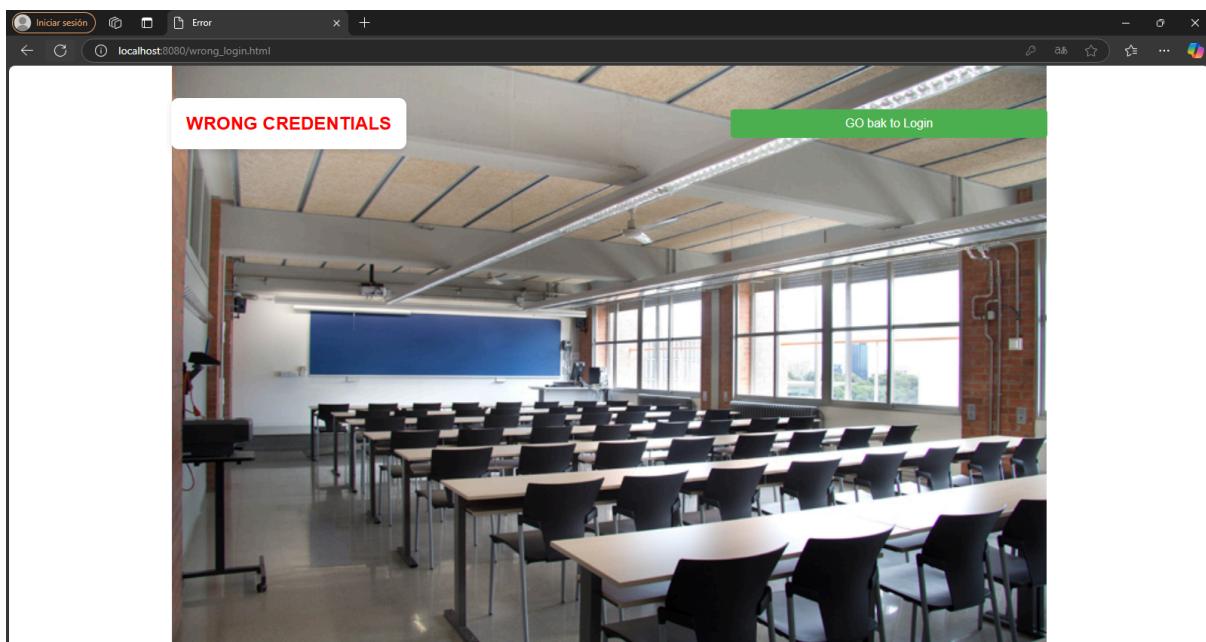
Si no tocamos nada en 50 segundo vuelve a la pantalla de login.



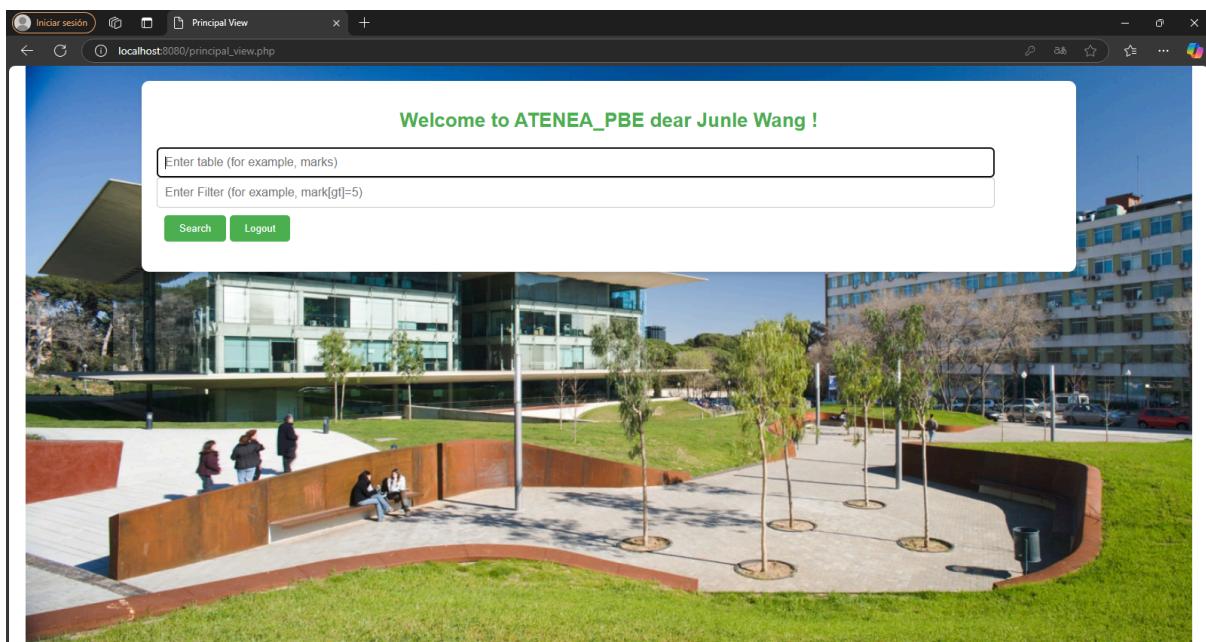
Este es el resultado de cuando entras en la página web.



Si nos dejamos un clave de acceso en blanco y le damos a login, nos notifica de que tenemos que rellenar ese campo.



Si ponemos mal el usuario o contraseña, nos muestra en la página **WRONG CREDENTIALS**, pero podemos volver a la página de inicio, dándole al botón verde que se muestra en pantalla.



Welcome to ATNEA_PBE dear Junle Wang !

Si rellenamos los siguientes campos así, nos mostrara la siguiente tabla

Welcome to ATNEA_PBE dear Junle Wang !

Enter table (for example, marks)

Enter Filter (for example, mark[gt]=5)

[Search](#) [Logout](#)

uid	subject	name	mark
5F63B81E	DSBM	Examen Parcial	6.00
5F63B81E	PBE	Examen Parcial	8.70

Asi nos ahorraremos introducir el &

Welcome to ATNEA_PBE dear Junle Wang !

Table Error, try again please.

Enter table (for example, marks)

Enter Filter (for example, mark[gt]=5)

[Search](#) [Logout](#)

Si nos equivocamos introduciendo la tabla, sale lo siguiente.

Cuando vuelvas a introducir bien la tabla, el mensaje en rojo se borrará.

Si le das al botón de Logout nos redirigirá a la página de inicio de sesión.