Marc Muñoz Eloi Saballs Alejandro de Alvarado Anna Llamas Èric Lozano

Critical Design Review

Client

client.py

```python
import gi
import threading
import requests
import json
from gi.repository import Gtk, GLib, Gdk
import pynfc

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

class CampusVirtualClient(Gtk.Window):
    def __init__(self):
        super().__init__(title="Campus Virtual")
        self.set_default_size(600, 400)
        self.set_border_width(10)

        self.load_css("style.css")

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

        self.label_login = Gtk.Label(label="Aproxima al lector la vostra
targeta")
        self.label_login.get_style_context().add_class("login-label")
        self.login_frame.pack_start(self.label_login, True, True, 0)

        self.add(self.login_frame)

        self.nfc = pynfc.Nfc()
        self.start_nfc_thread()

    def load_css(self, css_file):
        css_provider = Gtk.CssProvider()
        css_provider.load_from_path(css_file)
        screen = Gdk.Screen.get_default()
        context = Gtk.StyleContext()
        context.add_provider_for_screen(
            screen, css_provider, Gtk.STYLE_PROVIDER_PRIORITY_APPLICATION
        )
```

```
    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:
            print("Esperant a llegir la targeta...")
            try:
                target = next(self.nfc.poll())
                uid = target.uid.decode('ascii').upper()
                GLib.idle_add(self.handle_card_detected, uid)
                break
            except StopIteration:
                pass

    def handle_card_detected(self, uid):
        self.label_login.set_text(f"UID detected: {uid}")
        threading.Thread(target=self.query_student_name, args=(uid,
self.handle_load_student_interface)).start()

    def query_student_name(self, uid, handler):
        student_url = f"http://172.20.10.6:8000/students?uid={uid}"
        try:
            response = requests.get(student_url)
            response.raise_for_status()
            data = response.json()
            name = data[0].get("name", "Unknown Student") if data and
isinstance(data, list) else "Unknown Student"
            GLib.idle_add(handler, name, uid)
        except requests.RequestException:
            GLib.idle_add(self.label_login.set_text, "Error: Unable to fetch
student name.")

    def handle_load_student_interface(self, name, uid):
        self.login_frame.destroy()
        self.student_frame = Gtk.Box(orientation=Gtk.Orientation.VERTICAL,
spacing=10)
        self.label_welcome = Gtk.Label(label=f"Welcome {name}")
        self.label_welcome.get_style_context().add_class("welcome-label")
        self.student_frame.pack_start(self.label_welcome, False, False, 0)

        self.option_var = Gtk.ComboBoxText()
        self.option_var.append_text("Timetables")
        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.get_style_context().add_class("show-button")
        self.display_button.connect("clicked",
self.handle_display_button_clicked, uid)
        self.student_frame.pack_start(self.display_button, False, False, 0)

        self.data_view = Gtk.Box()
        self.student_frame.pack_start(self.data_view, True, True, 0)

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

    def handle_display_button_clicked(self, button, uid):
        option = self.option_var.get_active_text().lower()
        for child in self.data_view.get_children():
            self.data_view.remove(child)
        threading.Thread(target=self.query_server, args=(option,
uid)).start()

    def query_server(self, option, uid):
        server_url = f"http://172.20.10.6:8000/{option}?uid={uid}"
        try:
            response = requests.get(server_url)
            response.raise_for_status()
            data = response.json()

            if option == "timetables":
                columns = ["day", "hour", "subject", "room"]
            elif option == "tasks":
                columns = ["date", "subject", "name"]
            elif option == "marks":
                columns = ["subject", "name", "mark"]
            else:
                raise ValueError("Invalid option selected")

            GLib.idle_add(self.handle_display_table, data, columns)

        except requests.RequestException as e:
            GLib.idle_add(self.handle_display_data, f"Error loading
{option}:\n{e}")
        except json.JSONDecodeError:
            GLib.idle_add(self.handle_display_data, f"Error: Server did not
return valid JSON.")

    def handle_display_table(self, data, columns):
        store = Gtk.ListStore(*(str for _ in columns))
        for entry in data:
```

```
            row = [entry.get(col, "") for col in columns]
            store.append(row)
        treeview = Gtk.TreeView(model=store)
        for i, column_title in enumerate(columns):
            renderer = Gtk.CellRendererText()
            column = Gtk.TreeViewColumn(column_title.capitalize(), renderer,
text=i)
            treeview.append_column(column)
        scrolled_window = Gtk.ScrolledWindow()
        scrolled_window.add(treeview)
        scrolled_window.set_min_content_height(200)
        self.data_view.pack_start(scrolled_window, True, True, 0)
        self.show_all()

    def handle_display_data(self, data):
        text_view = Gtk.TextView()
        text_buffer = Gtk.TextBuffer()
        text_view.set_buffer(text_buffer)
        text_buffer.set_text(data)
        self.data_view.pack_start(text_view, True, True, 0)
        self.show_all()

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

style.css

```
Unset
window {
    background-color: #d9effc;
}

.login-label {
    font-size: 20px;
    font-weight: bold;
    color: #000000;
}

.welcome-label {
    font-size: 24px;
    font-weight: bold;
    color: #333;
}

button.show-button {
    background-color: #4CAF50;
    color: white;
    padding: 10px;
    border-radius: 5px;
    font-size: 16px;
}

button.show-button:hover {
    background-color: #45a049;
}

combobox {
    font-size: 14px;
    background-color: #fff;
    border: 1px solid #ccc;
    padding: 5px;
}

treeview {
    font-size: 14px;
    color: #333;
}
```

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.

Hem tingut 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.

Vam voler arreglar el nostre cdr, així que vam modificar el nostre codi del client i vam fer-ho mitjançant threads i handlers.

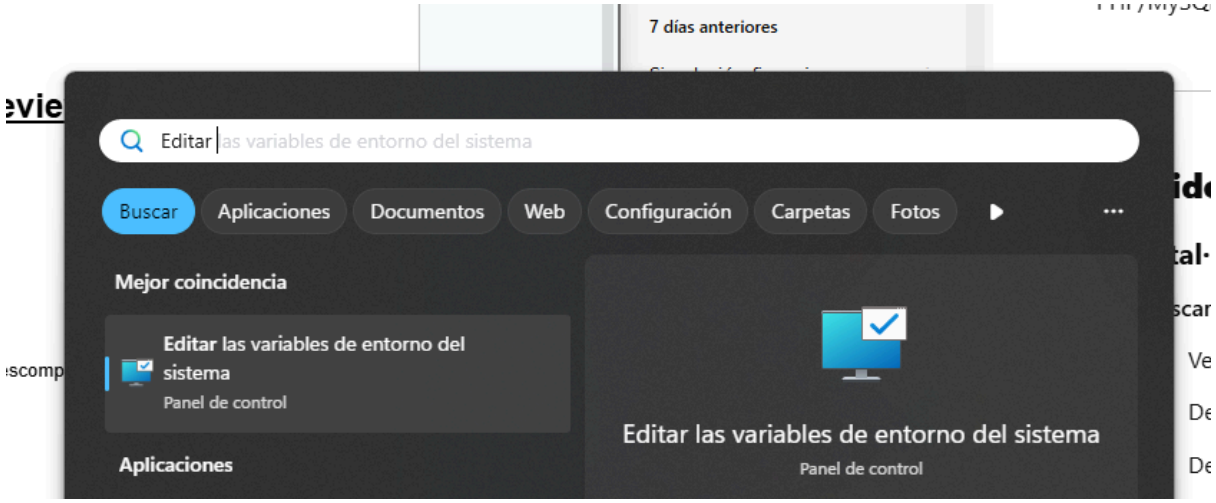
La gràcia de fer servir threads és evitar que les operacions de llarga duració (com ara l'accés al lector NFC o les consultes al servidor) bloquegin la interfície gràfica d'usuari (GUI).

Finalment, hem fet servir un handler, el qual s'encarrega dels events/resultats que succeeixen en segon pla, així la GUI es pot processar i actualitzar de manera segura i efectiva.

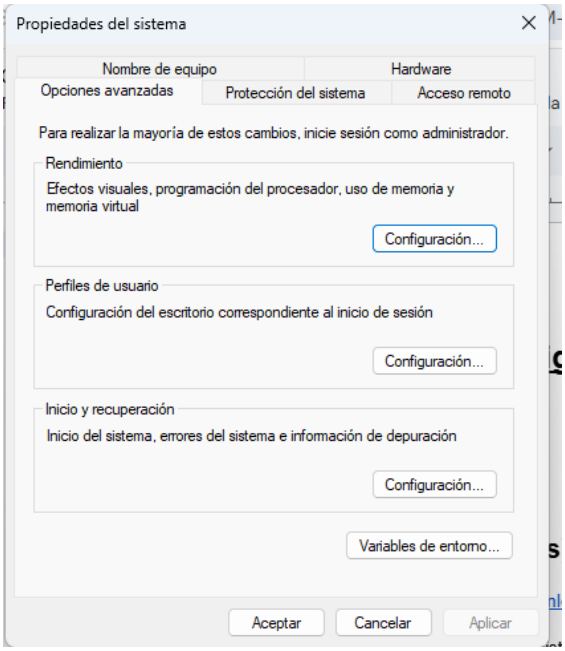
Servidor (Des de un Windows)

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

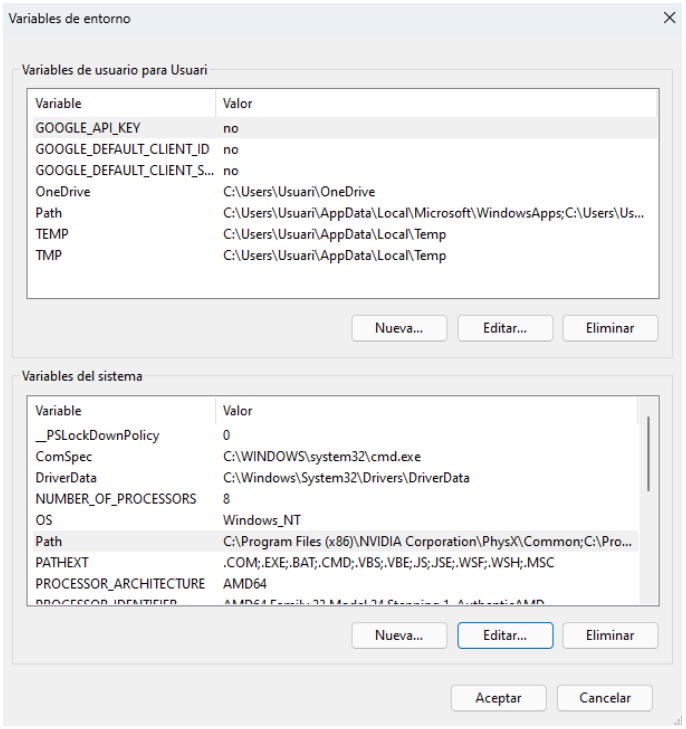
Configurar el path a les variables d'entorn del sistema:



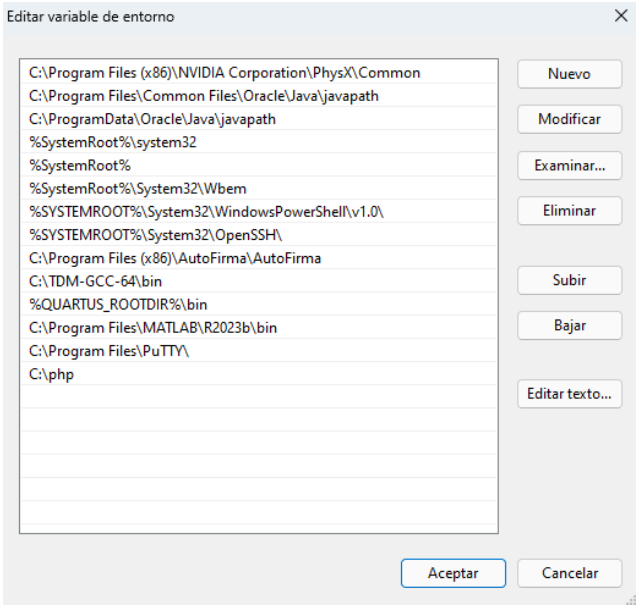
Entrar a "Editar las variables de entorno del sistema" i clicar a "Variables de entorno":



Editem les "Variables del sistema":



Escrivim “C:\php”:



Configurar el PHP: Obrir un terminal (php –ini), buscar a la carpeta de php el php.ini (en el nostre 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 descomprimirlo 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 inicializar 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 altre 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 altre 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 altre CMD (una altre 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

Index.php

```C/C++
<?php
$servername = "localhost";  //Parametres de la base de dades
$username = "root";
$password = "1234";
$dbname = "nemesis";

header("Content-Type: application/json");           //Capçalera per evitar el CORS error
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, OPTIONS");
header("Access-Control-Allow-Headers: Content-Type, Authorization");

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


if($_SERVER['REQUEST_METHOD'] === 'OPTIONS'){   //Quan demana el tipus OPTIONS que faci un redirecionament
    http_response_code(204);
    exit;
}

session_start();        //Inici de sessio
if (isset($_SESSION['last_activity']) && (time() - $_SESSION['last_activity'] > 900))
{     //15 minutos de inactividad
    session_unset();
    session_destroy();
    http_response_code(401);    //Error de client - Temps Maxim de inactivitat
    die(json_encode(["error" => "Sesion expirada."]));
}
$_SESSION['last_activity'] = time();

$uid = $_GET['uid'] ?? null;    //Validar UID
if (!$uid) {
    $username = $_GET['username'] ?? null;
    if (!$username) {
        http_response_code(402);    //Error de client - UID o Username no trobat
        die(json_encode(["error" => "UID no proporcionado."]));
    }
}

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

$request_uri = $_SERVER['REQUEST_URI'] ?? '';   //Separar la taula de las restriccions
$request_parts = explode('?', $request_uri, 2);     //Partir el string central a partir del '?'
$table = ltrim($request_parts[0], '/');     //La primera part es la taula
```

```php
$allowed_tables = ['marks', 'tasks', 'timetables', 'students'];     //Validar taula
if (!$table || !in_array($table, $allowed_tables)) {
    http_response_code(403);            //Error de Client - Taula donada invalida
    die(json_encode(["error" => "Tabla no valida."]));
}

$params = [];   //Inicialitzar la variable que contindra les restriccions
if (!empty($request_parts[1])) {    //La segona part son les restriccions
    parse_str($request_parts[1], $params);      //Dividim les restriccions
individualment en el params
}

$operator_map = [   //Mapejar els operadors
    'lt'  => '<',
    'lte' => '<=',
    'gt'  => '>',
    'gte' => '>=',
    'eq'  => '='
];

$constraints = [];  //Inicialitzacio de la variable que tindra els constraints
$order_by = "";     //Inicialitzacio de la variable que tindra el ordre
$limit = "";        //Inicialitzacio de la variable que tindra el limit

foreach ($params as $column => $conditions) {   //Analisis de cada condicio
    if (is_array($conditions)) {                //Separar las restricciones directas
        foreach ($conditions as $operator => $value) {
            if (isset($operator_map[$operator])) {      //Comprovar els operadors
                if ($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) . "'";   //Definicio del constrait
            }
        }
    } else {
        $constraints[] = $column . " = '" . $conn->real_escape_string($conditions) .
"'";   //Definicio del contrait en condicio directe
    }
}

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


switch ($table) {   //Ordre per defecte de cada taula
    case 'marks':
        $order_by = "ORDER BY subject";     //Taula marks - Ordre per subject
        break;
    case 'tasks':
        $order_by = "ORDER BY date";        //Taula tasks - Ordre per dia
```

```
        break;
    case 'timetables':                      //Taula timetables - Ordre per dia/hora a
partir de la actual
        $current_day = date('D');           //Formato: Mon, Tue, Wed...
        $current_time = date('H:i:s');      //Formato: HH:MM:SS
        $hora = explode(':', $current_time, 3);     //Separar la hora
        if($hora[1] > '00'){                        //Mirem si el minut es mes gran de
00
            $hora_actual = $hora[0] + '02';         //+2 Per truncar la hora
        } else {
            $hora_actual = $hora[0] + '01';         //+1 Per la hora desfassada
        }
        $days_map = [       //Mapejar els dies
            'Mon' => 1,
            'Tue' => 2,
            'Wed' => 3,
            'Thu' => 4,
            'Fri' => 5,
            'Sat' => 6,
            'Sun' => 7
        ];

        $current_day_index = $days_map[$current_day];       //Dia Actual (Numerico)
        $order_by = "ORDER BY
        CASE
            WHEN day = '$current_day' AND hour >= '$hora_actual' THEN 0
            WHEN day = '$current_day' AND hour < '$hora_actual' THEN 7
            ELSE (
                CASE day
                    WHEN 'Mon' THEN (7 + 1 - $days_map[$current_day]) % 7
                    WHEN 'Tue' THEN (7 + 2 - $days_map[$current_day]) % 7
                    WHEN 'Wed' THEN (7 + 3 - $days_map[$current_day]) % 7
                    WHEN 'Thu' THEN (7 + 4 - $days_map[$current_day]) % 7
                    WHEN 'Fri' THEN (7 + 5 - $days_map[$current_day]) % 7
                    WHEN 'Sat' THEN (7 + 6 - $days_map[$current_day]) % 7
                    WHEN 'Sun' THEN (7 + 7 - $days_map[$current_day]) % 7
                END
            )
        END,
        hour";
        break;
    case 'students':    //Taula students - No s'ordena
        break;
}

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

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

if (!empty($constraints)) {     //Comprobar si hi han restriccions
    $constraints = array_filter($constraints, function($constraint) {
```

```
        return !str_contains($constraint, 'limit');         //Eliminar 'limit' de les
restriccions
    });

    if (!empty($constraints)) {
        $sql .= " WHERE " . implode(" AND ", $constraints); //Afegir les restriccions
que encara no s'han afegit
    }
}

$sql .= " $order_by";   //Implementar el ORDER

$sql .= " $limit";      //Implementar el LIMIT

//echo "SQL Query: " . $sql . "\n";     //Per poder veure per pantalla la consulta
realitzada

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

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

$data = [];
while ($row = $result->fetch_assoc()) {     //Recull les dades
    $data[] = $row;
}

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

$conn->close();
?>
```

Aquest codi PHP implementa una API REST per interactuar amb una base de dades MySQL. Permet consultar dades de taules específiques (`marks`, `tasks`, `timetables`, `students`) aplicant restriccions i límits configurables a través de paràmetres de la URL. Gestiona sessions d'usuari, incloent-hi la caducitat per inactivitat, i retorna els resultats en format JSON. També inclou validacions per evitar errors de connexió, taules no vàlides i restriccions incorrectes.

Index.html

```
C/C++
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>NEMESIS UPC</title>
    <link rel="icon" href="FOTOS/Logo-Nemesis.jpg" type="image/x-icon">
    <!-- Codi CSS -->
    <link rel="stylesheet" href="CSS/styles.css">
</head>
<body>
    <!-- Foto del campus-->
    <div id="campus-photo" class="campus-foto">
        <img class="campus-nord" src="FOTOS/campus-nord.jpg" alt="Campus Nord">
    </div>
    <!-- Login Container -->
    <div id="login-container" class="centered">
        <img class="logo_upc" src="FOTOS/logo_upc.png" alt="Logo UPC">
        <form id="login-form">
            <input class="username" placeholder="Usuari" type="text" id="username"
name="username" autocomplete="username" required>
            <input class="pssw" placeholder="Contrasenya" type="password"
id="password" name="password" autocomplete="current-password" required>
            <button class="login-button" type="submit">Entra</button>
        </form>
        <p id="error-message" class="error"></p>
    </div>
    <!-- Pagina Principal (Inicialment Ocult)-->
    <div id="dashboard-container" class="hidden">
        <!-- Header de la Pagina Principal-->
        <header class="dashboard-header">
            <img class="logo_upc-small" src="FOTOS/Logo-sense-fons.png" alt="Logo
NEMESIS">
            <p id="welcome-message">Welcome, <span id="user-name"></span></p>
            <button id="logout-button" class="logout-button">Logout</button>
        </header>
        <!-- Contingut Principal -->
        <div id="main-content">
            <div id="search-section">
                <div class="dropdown">
                    <select id="table-selector">
                        <option value="">Select Table</option>
                        <option value="marks">Marks</option>
                        <option value="tasks">Tasks</option>
                        <option value="timetables">Timetables</option>
                    </select>
                </div>
                <input type="text" id="query" placeholder="Enter your query">
                <button id="search-button">Send</button>
            </div>
            <!-- Taula de Resultats-->
            <div id="results-container"></div>
```

```
        </div>
    </div>
    <!-- Codi JavaScript-->
    <script src="JS/main.js"></script>
</body>
</html>
```

Aquest codi HTML estructura una pàgina web interactiva que inclou les següents funcionalitats:

1. **Portada amb login:** Mostra una imatge del Campus Nord i un formulari de login on els usuaris poden introduir el seu nom d'usuari i contrasenya.
2. **Icones i estils personalitzats:** Utilitza fitxers externs per a l'estil (CSS) i el comportament (JavaScript).
3. **Dashboard ocult inicialment:** Després del login, es mostra un panell principal amb funcions com la selecció de taules (marks, tasks, timetables) i la realització de consultes.
4. **Disseny responsive i estructurat:** S'assegura que el contingut sigui accessible en diferents dispositius gràcies a la metainformació i les classes CSS.
5. **Integració de JavaScript:** El fitxer `main.js` proporciona la lògica per gestionar les interaccions, com ara el login i la cerca.

main.js

JavaScript

```
const API_URL = "http://192.168.1.74:8000";
//URL del servidor PHP

document.getElementById("login-form").addEventListener("submit", async function
(event) {                                   //Quan s'envia el submit del formulari del
login
event.preventDefault();
//Evita que es refresqui la pestanya

    const username = document.getElementById("username").value;
//Obtenir els valors de username
    const password = document.getElementById("password").value;
//i password

    try {
        const userData = await login(username, password);
//Executa la funcio login i guardem a userData el seu return
        if (userData) {
//Si la hi ha algo a la userData
            displayDashboard(userData);
//Mostra el dashboard (Pantalla inicial)
        }
    } catch (error) {
//En cas de haver algun problema amb el inici de funcio es retorna el error adient
        document.getElementById("error-message").innerText = error;
    }
});

async function login(username, password) {
//Funcio encarregada de verificar el username i password
    const url =
`${API_URL}/students?username=${encodeURIComponent(username)}&password=${encodeURIComp
onent(password)}`;    //Creem la url que contindra la pregunta de forma adequada
perque el servidor PHP pugui entendre
    const response = await fetch(url, { method: "GET" });
//Guardem la resposta a response

    if (!response.ok) {
//En cas de que la resposta no estigui correcte
        throw new Error("Login failed");
//Enviar Error - Login incorrecte
    }

    const userDataArray = await response.json();
//Processa la resposta .JSON

    if (userDataArray.length === 0) {
        throw new Error("Invalid username or password");
//Enviar Error - Username o Password incorrecte
    }
```

```
    const userData = userDataArray[0];
//Ens quedem el primer element del array (El nostre usuari)

    return userData;
//Retorna el userData del usuari
}

function displayDashboard(userData) {
//Mostrar dashboard despres de fer un login correcte
    document.getElementById("login-container").classList.add("hidden");
//Amaga el login container
    document.getElementById("dashboard-container").classList.remove("hidden");
//Apareix el dashboard container
    document.getElementById("user-name").textContent = userData.name;
//Donem el nom de usuari al text de Welcome
    const campusPhoto = document.getElementById("campus-photo");
    if (campusPhoto) {
//Ocultem la foto del campus
        campusPhoto.classList.add("hidden");
        campusPhoto.style.visibility = 'hidden';
//No reservar espai per aquest
        campusPhoto.style.display = 'none';
//No ocupar espai en el layout
    }
    window.currentUser = userData;
//Guarda les dades del usuario
}

document.getElementById("search-button").addEventListener("click", async function () {
//Quan es clica Search
    const query = document.getElementById("query").value.toLowerCase();
//Busca el query
    const studentId = window.currentUser.uid;
//Agafem la uid del currentUser que hem guardat abans
    const table = document.getElementById("table-selector").value;
//Busca la taula selecionada en la finestra desplegable

    console.log("Selected table:", table);
// Verificar el valor en consola
    if (!table) {
        document.getElementById("results-container").innerText = "Please select a
table.";                                //Si no es troba una taula seleccionada
enviar el error
        return;
//Sortir i no executar res mes.
    }

    try {
        const data = await fetchTableData(table, studentId, query);
//Funcio per enviar la query de dades
        displayTable(data);
//Funcio per ensenyar les dades en una taula
    } catch (error) {
```

```
        document.getElementById("results-container").innerText = `Error loading data
for.` + error;                          //Si es detecta un error es mostra en lloc dels
resultats
    }
});

async function fetchTableData(table, studentId, query) {
//Funcio que donad la taula, el uid i la query, envia la URL i retorna les dades
resultants
    const url =
`${API_URL}/${encodeURIComponent(table)}?uid=${encodeURIComponent(studentId)}&${(query
)}`;                  //Creacio de la URL

    const response = await fetch(url, { method: "GET" });
//Espera la resposta

    return await response.json();
//Retorna la resposta en format JSON
}

function displayTable(data) {
//Funcio que crea i emplena les taules de dades
    const container = document.getElementById("results-container");
//Adquereix la variable del dashboard on volem la taula
    container.innerHTML = "";
//S'assegura que estigui buit

    if (Array.isArray(data) && data.length > 0) {
//S'assegura que les dades son un array i es no null
        const table = document.createElement("table");
//Creem la taula 'table'
        table.border = "1";

        const thead = document.createElement("thead");
//Creem el table header 'thead'
        const headerRow = document.createElement("tr");
//Creem el headerRow

        const keys = Object.keys(data[0]).filter((key) => key !== 'uid');
//Emplenem la variable keys
        keys.forEach((key) => {
//Per cada key crea un th amb el seu valor i afegeix los a headerRow
            const th = document.createElement("th");
            th.textContent = key.toUpperCase();
            headerRow.appendChild(th);
        });

        thead.appendChild(headerRow);
//Afegeix headerRow a thead
        table.appendChild(thead);
//Afegeix thead a table

        const tbody = document.createElement("tbody");
//Crear filas de datos
```

```
        data.forEach((row) => {
//Per cada fila crea un tr (table row)
            const tr = document.createElement("tr");
            keys.forEach((key) => {
//Per cada key de cada fila crea un td amb el valor pertinent
                const td = document.createElement("td");
                td.textContent = row[key];
                tr.appendChild(td);
//Afegeix els td al tr de cada fila
            });
            tbody.appendChild(tr);
//Afegeix el tr de cada fila a el tbody
        });

        table.appendChild(tbody);
//Afegeix el tbody a la table
        container.appendChild(table);
//Afegeix la table al container principal

    } else {
//En cas que dades no sigui un array o sigui null
        const th = document.createElement("th");
        th.textContent = "No data available";
//Apareix aquest error
        container.appendChild(th);
//Afegeix el error en el container principal
    }
}

document.getElementById("logout-button").addEventListener("click", function () {
//Logout button (tornar a la login-form)
    window.currentUser = null;
//Eliminem les dades del user que estava utilitzant la sessio
    document.getElementById("dashboard-container").classList.add("hidden");
//Amaga el dashboard container
    document.getElementById("login-container").classList.remove("hidden");
//Apareix el login container
    document.getElementById("query").value = "";
//Resetejem el query
    document.getElementById("table-selector").value = "";
//Resetejem la finestra desplagable
    document.getElementById("results-container").innerHTML = "";
//Resetejem el results container
    const campusPhoto = document.getElementById("campus-photo");
    if (campusPhoto) {
//Resetejar els valors de la imatge perque es torni a veure com al principi
        campusPhoto.classList.remove("hidden");
        campusPhoto.style.visibility = 'visible';
        campusPhoto.style.display = 'flex';
    }
});
```

Aquest codi JavaScript implementa la lògica principal d'una aplicació web que inclou autenticació i un sistema de consultes de dades.

1. **Login d'usuaris:** Gestiona l'autenticació enviant les credencials a un servidor PHP. Si l'inici de sessió té èxit, oculta el formulari de login i mostra un panell principal (dashboard).
2. **Interacció amb el servidor:** Realitza peticions GET per recuperar dades (com marks, tasks o timetables) basant-se en l'usuari autenticat, les taules seleccionades i les consultes introduïdes.
3. **Visualització de dades:** Crea dinàmicament taules HTML per mostrar els resultats de les consultes al panell principal o un missatge si no hi ha dades disponibles.
4. **Gestió de sessió:** Inclou una funcionalitat de tancament de sessió que restableix l'estat inicial de l'aplicació, incloent el formulari de login i la imatge del campus.
5. **Control d'errors:** Mostra missatges d'error per credencials incorrectes, problemes de connexió o consultes invàlides.

styles.css

```
Unset
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: white;
    color: #333;
    display: flex;
    justify-content: right;
    height: 100vh;
    overflow: hidden;
}

.centered {
    text-align: center;
    background: white;
    color: white;
    padding: 20px;
    width: 300px;
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
}

#campus-photo {
    background-image: url('FOTOS/campus-nord.jpg');
    background-size: cover;
    background-position: center center;
    height: 100vh;
    width: 100%;
    display: flex;
}


.logo_upc{
    width: 300px;
    margin-top: 30px;
    margin-bottom: 50px;
    margin-right: 5px;
    margin-left: 5px;

}

.login-button{
    border-radius: 50px;
    background-color: #0270d7;
    width: 300px;
    padding-top: 20px;
    padding-bottom: 20px;
    margin-top: 30px;
}

.campus-foto{
    display: grid;
    width:auto;
```

```
    position: relative;
}

.campus-foto.hidden {
    visibility: hidden;
    display: none;
    position: static;
}

.username,
.pssw {
    outline: none;
}

.hidden {
    display: none;
}

form {
    display: flex;
    flex-direction: column;
}

input, button {
    margin: 10px 0;
    padding: 10px;
    border-radius: 5px;
    border: none;
    font-size: 16px;
}

button {
    background-color: #0055aa;
    color: white;
    cursor: pointer;
}

button:hover {
    background-color: #004499;
}

.error {
    color: #ff4d4d;
    margin-top: 10px;
}

.dashboard-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 20px;
    background-color: #0270d7;
    color: white;
}
```

```
.dashboard-header .logo_upc-small {
    width: 100px;
    height: auto;
    z-index: 10;
    visibility: visible;
}

#welcome-message {
    font-size: 18px;
}


.logout-button {
    background-color: #ff4d4d;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

.logout-button:hover {
    background-color: #ff1a1a;
}

#main-content {
    padding: 20px;
    text-align: center;
}

#search-section {
    margin-bottom: 20px;
}


#table-selector, #query, #search-button {
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
    border: 1px solid #ccc;
}

#search-button {
    background-color: #0055aa;
    color: white;
    cursor: pointer;
}

#search-button:hover {
    background-color: #004499;
}

#error-message {
```

```
    color: #ff4d4d;
    margin-top: 10px;
}

#results-container {
    margin-top: 20px;
}

#dashboard-container {
    text-align: left;
    width: 100%;
    overflow: scroll;
}

#results-container {
    margin-top: 20px;
    display: flex;
    justify-content: center;
    align-items: center;
}

table {
    border-collapse: collapse;
    width: 80%;
    margin: 20px 0;
}

th, td {
    padding: 10px;
    text-align: center;
    border: 1px solid #ccc;
}

th {
    background-color: #f4f4f4;
}
```

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 nemesis;
USE nemesis;

-- Crear la tabla IF NOT EXISTS Timetables
CREATE TABLE Timetables (
    day VARCHAR(10),
    hour INT,
    subject VARCHAR(50),
    room VARCHAR(10),
    uid VARCHAR(10)
);

-- Crear la tabla Marks
CREATE TABLE IF NOT EXISTS Marks (
    subject VARCHAR(50),
    name VARCHAR(50),
    mark DECIMAL(4, 2),
    uid VARCHAR(10)
);

-- Crear la tabla Tasks
CREATE TABLE IF NOT EXISTS Tasks (
    date DATE,
    subject VARCHAR(50),
    name VARCHAR(50),
    uid VARCHAR(10)
);

-- Crear la tabla Students
CREATE TABLE IF NOT EXISTS Students (
    uid VARCHAR(10),
    name VARCHAR(100),
    username VARCHAR(100),
    password VARCHAR(50)
);
```

Aquest codi inclou les instruccions bàsiques per crear les taules. posteriorment, s'ha d'anar afegint la informació i les dades corresponents a cada una de les taules, de manera que quedi com es mostra a continuació:

Taula d’estudiants:

```
mysql> SELECT * FROM students;
+----------+----------------------+----------------------+----------+
| uid      | name                 | username             | password |
+----------+----------------------+----------------------+----------+
| 06F3E0B0 | Alejandro de Alvarado | alejandro.de.alvarado | 1234     |
| 13503125 | Marc Muñoz           | marc.muñoz           | 1234     |
| 169C9314 | Èric Lozano          | eric.lozano          | 1234     |
| 66269414 | Anna Llamas          | anna.llamas          | 1234     |
| A2198D27 | Eloi Saballs         | eloi.saballs         | 1234     |
+----------+----------------------+----------------------+----------+
5 rows in set (0.03 sec)
```

Taula de notes:

```
mysql> SELECT * FROM marks;
+---------+---------------+-------+----------+
| subject | name          | mark  | uid      |
+---------+---------------+-------+----------+
| EIM     | Examen Parcial |  8.40 | A2198D27 |
| TCGI    | Examen Final   |  7.10 | A2198D27 |
| PBE     | CDR            |  8.60 | A2198D27 |
| PSAVC   | Examen Final   |  5.60 | A2198D27 |
| RP      | Examen Lab     |  9.30 | A2198D27 |
| CAL     | Examen Parcial |  1.50 | 169C9314 |
| ALG     | Examen Parcial |  0.20 | 169C9314 |
| FDF     | Examen Parcial |  5.10 | 169C9314 |
| FDE     | Examen Parcial |  7.50 | 169C9314 |
| FO      | Examen Parcial |  9.70 | 169C9314 |
| TFG     | Entrega Final  | 10.00 | 169C9314 |
| CAL     | Examen Final   |  9.20 | 13503125 |
| EM      | Examen Parcial |  3.10 | 13503125 |
| FISE    | Examen Parcial |  7.50 | 13503125 |
| RP      | Examen Parcial |  4.30 | 13503125 |
| EIM     | Examen Final   |  9.60 | 13503125 |
| FDE     | Examen Parcial |  8.20 | 66269414 |
| ALG     | Examen Final   |  3.40 | 66269414 |
| FO      | Examen Final   | 10.00 | 66269414 |
| ENTIC   | Projecte Final |  9.10 | 66269414 |
| POO     | Projecte Final |  6.50 | 66269414 |
| POO     | Projecte       |  9.90 | 06F3E0B0 |
| PIE     | Examen Final   |  4.70 | 06F3E0B0 |
| IPAV    | Examen Final   |  9.70 | 06F3E0B0 |
| ICOM    | Examen Parcial |  6.50 | 06F3E0B0 |
| DSBM    | Examen Lab     |  1.80 | 06F3E0B0 |
+---------+---------------+-------+----------+
26 rows in set (0.00 sec)
```

Taula de tasques:

```
mysql> SELECT * FROM tasks;
+------------+---------+---------------+----------+
| date       | subject | name          | uid      |
+------------+---------+---------------+----------+
| 2025-01-07 | PIE     | Examen Final  | 06F3E0B0 |
| 2025-01-08 | POO     | Examen Final  | 06F3E0B0 |
| 2025-01-10 | ICOM    | Examen Final  | 06F3E0B0 |
| 2025-01-15 | DSBM    | Examen Final  | 06F3E0B0 |
| 2025-01-20 | IPAV    | Examen Final  | 06F3E0B0 |
| 2025-01-08 | POO     | Examen Final  | 66269414 |
| 2025-01-09 | FDE     | Examen Final  | 66269414 |
| 2025-01-11 | ENTIC   | Entrega Final | 66269414 |
| 2025-01-11 | AL      | Examen Final  | 66269414 |
| 2025-01-17 | FO      | Examen Final  | 66269414 |
| 2025-01-07 | RP      | Examen Final  | 13503125 |
| 2025-01-07 | CAL     | Examen Final  | 13503125 |
| 2025-01-08 | FISE    | Examen Final  | 13503125 |
| 2025-01-14 | EIM     | Examen Final  | 13503125 |
| 2025-01-15 | EM      | Examen Final  | 13503125 |
| 2025-01-07 | CAL     | Examen Final  | 169C9314 |
| 2025-01-09 | FDE     | Examen Final  | 169C9314 |
| 2025-01-13 | AL      | Examen Final  | 169C9314 |
| 2025-01-15 | FDF     | Examen Final  | 169C9314 |
| 2025-01-17 | FO      | Examen Final  | 169C9314 |
| 2025-01-07 | RP      | Examen Final  | A2198D27 |
| 2025-01-09 | PSAVC   | Examen Final  | A2198D27 |
| 2025-01-13 | PBE     | Examen Final  | A2198D27 |
| 2025-01-14 | EIM     | Examen Final  | A2198D27 |
| 2025-01-16 | TCGI    | Examen Final  | A2198D27 |
+------------+---------+---------------+----------+
25 rows in set (0.01 sec)
```

Taula d’horaris:

day	hour	subject	room	uid
Mon	10	RP	A4105	A2198D27
Mon	14	EIM	A3201	A2198D27
Tue	8	PSAVC	A4105	A2198D27
Wed	8	Lab PBE	A4105	A2198D27
Wed	14	EIM	A3201	A2198D27
Wed	17	TCGI	A3201	A2198D27
Thu	8	PBE	A4105	A2198D27
Thu	10	RP	A4105	A2198D27
Thu	12	Lab RP	D3006	A2198D27
Thu	14	TCGI	A3201	A2198D27
Thu	16	Lab TCGI	A2S105	A2198D27
Fri	10	PSAVC	A4105	A2198D27
Mon	9	CAL	A3102	169C9314
Mon	11	AL	A3102	169C9314
Mon	12	Lab FO	A2S109	169C9314
Tue	8	FDF	A3102	169C9314
Tue	10	Lab FDE	C4S102	169C9314
Tue	12	FO	A3102	169C9314
Wed	8	AL	A3102	169C9314
Wed	10	CAL	A3102	169C9314
Thu	8	CAL	A3102	169C9314