

Trabajo previo práctica 3 PSCD  
Pablo Augusto Delgado 842255 y Miguel Aréjula Aisa 850068

```
#include <iostream>
#include "Semaphore_V4.hpp"

using namespace std;

const int N_EST = 60; // de estudiantes
const int N_FIL = N_EST/2; // de filas en la matriz
const int N_COL = 1000; // de columnas

bool examen_fin[N_EST] = {false};
bool estu_fin[N_FIL][2] = {false};
int silla = 0; // 0 = 0 sillas ocupadas , 1 = una silla ocupada, 2 = 2 sillas ocupadas
int silla1; // su valor es el del estudiante sentado
int silla2; // su valor es el del estudiante sentado
int resultado[N_EST];

//-----
//Pre: <fila> es un índice de fila de <D>
//Post: devuelve el máximo de la fila <fila>
int maxFila(int D[N_FIL][N_COL],int fila) {
    int max;
    return max;
}

//Pre: <fila> es un índice de fila de <D>
//Post: devuelve la suma de los els. de la fila <fila>
int sumaFila(int D[N_FIL][N_COL],int fila) {
    int sum;
    return sum;
}

//-----
void Estudiante(int nip ,int fila, bool hayFila,int miPareja, int D[N_FIL][N_COL], int silla, int
resultado[N_EST], int fin_examen[N_EST]) {

    // esperar por una silla libre
    <await (silla<2) > // una de las dos esté libre
    //esperar me sea asignada pareja y fila
    <await (hayFila = true && pareja[nip]>=0 )
        int miFila = fila;
        int miPareja = pareja[nip];
        silla = 0; >
```

```

        if (nip<miPareja) {
            // calcular máx de mi fila
            int resultado[nip] = maxFila();
            //hacérselo llegar a mi pareja
            fin_examen[nip]=true
        }
        else {

            // calcular la suma de mi fila
            int resultado[nip] = sumaFila();
            //coger info de max (de mi pareja)
            <await(fin_examen[miPareja]= true)
                //mostrar resultados
                //comunicar finalización
                fin_examen[fila]=true;>

            //-----
        }
    }
}

void Profesor (int silla, int silla1, int silla2, int pareja[N_EST], int fila) {
    for(int i=0; i<N_FIL; i++) {
        // esperar a que haya dos
        <await silla = 2>

        //comunicar a cada uno su pareja, y la fila que les toca
        pareja[silla1]= silla2;
        pareja[silla2] = silla1;
        fila = i;
    }
    // esperar que todos hayan terminado
    <await terminado true
        fin examen >
}

int main(){
    int D[N_FIL][N_COL]; //para almacenar los datos
    int fila = 0; //cada pareja cogerá una
    int pareja[N_EST]; //pareja[i] será la pareja asignada

    //cargar "datos.txt" en "D"
    ...
    cout << "Prueba finalizada\n";
    return 0;
}

```