

TRABAJO PRÁCTICO INTEGRADOR

PARTE 1

Docente: Sanroman, Gabriela

Alumnos: Gomez, Sol; Marquez, Bautista; Varela, Lucas;
Vasques, Kalil

Materia: Algoritmos y Estructuras de Datos

Comisión: K1022

Institución: Universidad Tecnológica Nacional

Ciclo Lectivo: 2022

¿Cómo se usa?

Se despliega un menú que le solicita la información pertinente al usuario, indicándole cómo finalizar el ingreso de datos.

```
Si quiere finalizar el ingreso de datos, presione x

Ingrese el idioma que desee, sin tilde por favor: ingles
Ingrese el nivel: 4
Ingrese codigo de curso: 1022
Ingrese cupo del curso: 25
Ingrese nombre del docente a cargo: alejandro
Ingrese documento del docente a cargo: 44829165

-----

Ingrese el idioma que desee, sin tilde por favor:
```

Si se ingresa mal el nivel, el programa lo detecta y pide la corrección. Esto permite evitar errores.

```
Si quiere finalizar el ingreso de datos, presione x

Ingrese el idioma que desee, sin tilde por favor: ingles
Ingrese el nivel: 1022
Por favor, ingrese un numero valido del 1 al 8 :)
4
Ingrese codigo de curso: 1022
Ingrese cupo del curso: 24
Ingrese nombre del docente a cargo: alejandro
Ingrese documento del docente a cargo: 44829165

-----
```

Cuando se finaliza el ingreso de datos, se muestra por consola la cantidad de cursos por cada idioma. Especificando, además, si hay algún idioma con cursos en todos los niveles, o si hay alguno de los 8 niveles sin idiomas.

```
-----  
Ingrese el idioma que desee, sin tilde por favor:
```

```
x
```

```
Idiomas con cursos en todos los niveles:
```

```
Ingles
```

```
Niveles sin idiomas:
```

```
NINGUNO
```

```
  Cursos de Ingles: 8
```

```
Cursos de Portugues: 1
```

```
Cursos de Italiano: 1
```

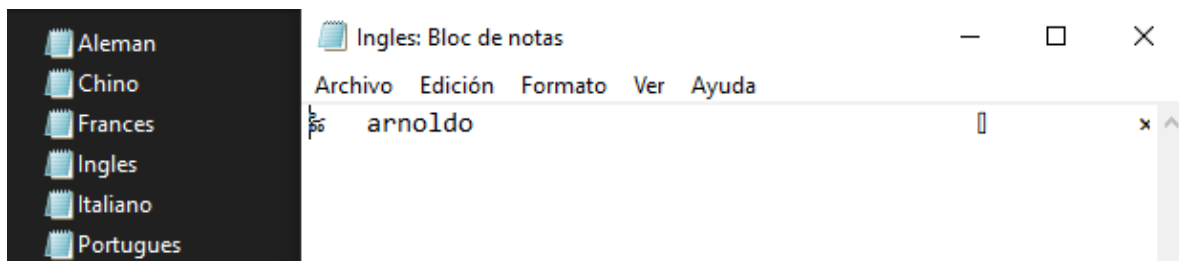
```
Cursos de Chino: 2
```

```
Cursos de Aleman: 1
```

```
Cursos de Frances: 1
```

Así se ve el código después de ser ejecutado

Se genera 1 archivo .dat por cada idioma. Al ser un archivo binario no es directamente legible.



Explicación del código

El menú que le da vida al programa.

Registra el idioma ingresado y verifica si coincide con alguno de los 6 disponibles, en caso afirmativo guarda un valor entero dependiendo del idioma en cuestión, y ese es el dato que pasa a la función **CargarDatos**.

```

do {
    int cont = 0;
    for (int i = 0; i < 6; i++) {
        if (idiomaIngresado != idiomas[i] && idiomaIngresado != idiomas2[i] && idiomaIngresado != idiomas3[i]) {
            cont++;
        }
    }
    if (cont == 6) {
        cout << "Ingrese un idioma valido, por favor: ";
        cin >> idiomaIngresado;
    }
} //CREAR ARCHIVOS
for (int ac = 0; ac < 6; ac++) {
    CrearArchivos(n: ac, x: idiomas);
}
//Selección de idioma
for (int i = 0; i < 6; i++) {
    if (idiomaIngresado == idiomas[i] || idiomaIngresado == idiomas2[i] || idiomaIngresado == idiomas3[i]) {
        NIdioma = i;
        contador[i]++;
    }
}

//Asocio el nivel al parametro y posicion del array de ese idioma
switch (NIdioma) {...}
cout << "Ingrese el idioma que desee, sin tilde por favor: ";
cin >> idiomaIngresado;

} while (idiomaIngresado != "x" && idiomaIngresado != "X");

```

La función **CargarDatos** es la que se encarga de pedir la información al usuario. Lo que hace es pedir los valores que faltan, verificar que el nivel, cupo y código estén en sus respectivos rangos aceptables y en caso de necesitar corrección, solicitarla. Esto para después guardar esa información en los parámetros asociados a la posición correspondiente del arreglo de cursos.

```

void CargarDatos(Instituto idioma[], int contador2[]){
    int ingresoNivel;

    cout << "Ingrese el nivel: "; //Ingreso de datos por consola
    cin >> ingresoNivel;
    while(ingresoNivel < 0 || ingresoNivel > 9){
        cout << "Por favor, ingrese un numero valido del 1 al 8 :)\n";
        cin >> ingresoNivel;
    }
    contador2[ingresoNivel-1]++;

    while(idioma[ingresoNivel-1].codigoDeCurso != -1){
        cout << "Curso existente, ingrese otro: ";
        cin >> ingresoNivel;
    }
    cout << "Ingrese codigo de curso: ";
    cin >> idioma[ingresoNivel-1].codigoDeCurso;

    while(idioma[ingresoNivel-1].codigoDeCurso < 1){
        cout << "Codigo incorrecto, ingrese codigo de curso nuevamente: ";
        cin >> idioma[ingresoNivel-1].codigoDeCurso;
    }

    cout << "Ingrese cupo del curso: ";
    cin >> idioma[ingresoNivel-1].cupos;

    while(idioma[ingresoNivel-1].cupos < 1){
        cout << "Ingrese cupo de curso nuevamente: ";
        cin >> idioma[ingresoNivel-1].cupos;
    }
    cout << "Ingrese nombre del docente a cargo: ";
    cin >> idioma[ingresoNivel-1].nombre;

    cout << "Ingrese documento del docente a cargo: ";
    cin >> idioma[ingresoNivel-1].dni;
    if(idioma[ingresoNivel-1].dni < 1){
        cout << "Ingrese documento del docente a cargo: ";
        cin >> idioma[ingresoNivel-1].dni;
    }

    cout << "\n-----\n" << endl;
}

```

Para informar la cantidad de niveles por idioma se verifica primero si existe algún idioma con 8 niveles. Esto se prueba para cada uno de los idiomas con un contador pasado como parámetro (que incrementa en 1 cada vez que se agrega un curso a ese idioma).

Si ningún idioma tiene 8 niveles, se devuelve esa información por consola.

En caso de que algún nivel no tuviera idiomas (esto se recorre al revés), la variable ex2 (de existencia) no incrementaría nunca y nos devolvería por consola la no existencia de lo buscado.

```

void Informar(int contadorI[], string idiomasI[], int contador2I[]){
    int ex=0, ex2=0;

    cout << "Idiomas con cursos en todos los niveles:\n";

    for(int info=0; info<6 ; info++){
        if(contadorI[info] == 8){
            cout << idiomasI[info] << endl;
            ex++;
        }
    }
    if(ex == 0){
        cout << "\nNINGUNO\n";
    }

    cout << "\n\nNiveles sin idiomas:\n";

    for(int info2=0; info2<8; info2++){
        if(contador2I[info2] == 0){
            cout << info2 + 1 << endl;
            ex2++;
        }
    }
    if(ex2 == 0){
        cout << "NINGUNO\n ";
    }
}

```

EscribirArchivos es la función responsable de cargar la información en los mismos. Lo que hace es crear un .dat, escribiendo en este la información almacenada en el array Instituto (ordenando por código de curso). La función se llama 1 vez por cada idioma que se quiera crear.

```

void EscribirArchivos(Instituto v[] , int contadorF[], int posicionF, string nombreArch){
    string nombre = nombreArch + ".dat";
    f = fopen( Filename: nombre.c_str(), Mode: "wb");

    for(int aa=0 ; aa < contadorF[posicionF] ; aa++){//
        fwrite( Str: &v[aa], Size: sizeof(Instituto), Count: 1, File: f);
    }
    fclose( File: f);
}

```