

Número y nombre del curso	8-5 IFCT087PO. PROGRAMACIÓN VISUAL C++
---------------------------	--

Nombre y apellidos del alumno	ADRIAN SEBASTIAN PAUSENIUC CIARNAU
NIF del alumno	60247639J
Fecha de la prueba	18/04/2024
Firma	

INSTRUCCIONES DE LA PRUEBA

Examen teórico práctico con 5 preguntas tipo test y 1 ejercicio práctico.
 Cada pregunta correcta suma 1 punto.
 El ejercicio suma 5 pts.
 Para aprobar es necesario obtener al menos 5 puntos en el conjunto del examen.
 Sube este documento marcando tus respuestas o con el texto Respuesta: *respuesta* al lado de la pregunta.
 Firma el documento y guárdalo sin protección de escritura.
 Súelo en .zip al Campu,s junto con el ejercicio práctico.

PREGUNTAS RESUELTAS

1. Concatenar

¿Cómo harías para concatenar dos valores?

- ☒ a. Con el signo +
- ☐ b. Con el signo &
- ☐ c. Con el signo ""
- ☐ d. con el signo >>

2. ARRAYS

```
int num[6] = {2,6,3,1,4,5};
```

¿Qué salida daría la siguiente declaración?

```
num[3];
```

- ☐ a. 4
- ☒ b. 1
- ☐ c. 2
- ☐ d. 5
- ☐ e. 6
- ☐ f. 3

3. OBJETOS

¿Cuál es la forma correcta de crear un objeto?

- ☐ a. `class MiNombreObjeto = new(MiNombreClase) ;`
- ☒ b. `MinombreObjeto class();`
- ☐ c. `miNombreClase MinombreObjeto;`
- ☐ d. `class.miNombreClase = new(MinombreObjeto);`

4. Manejo de archivos

¿Qué debemos incluir para poder manejar archivos?

- ☐ a. `#include <isostream>`
- ☐ b. `#include <ifstream>`
- ☒ c. `#include <fstream>`

5. SQL

¿Cuál es la sintaxis correcta para esta consulta sql?

- ☐ a. `UPDATE album SET (album_id==1);`
- ☒ b. `UPDATE album SET album_id=1;`
- ☐ c. `UPDATE INTO album SET album_id==1;`
- ☐ d. `UPDATE album SET (album_id,1);`

6. EJERCICIO PRÁCTICO

Crea un proyecto para una aplicación de dos ventanas que contenga, la primera, un formulario de login con email y password. Valida los datos con los siguientes: email: email@email.com - password: password123, y si son correctos, pasa a la segunda ventana, si no, mostramos una ventana con un mensaje de error (MessageBox). En la segunda ventana dibuja un gridview que se alimente del siguiente Array.

```
array<Alumnos^>^ alumnosArray = gcnnew array<Alumnos^> {
    gcnnew Alumnos { "Juan", 25 },
    gcnnew Alumnos { "María", 30 },
    gcnnew Alumnos { "Carlos", 40 }
};
```

Deberás crear la clase alumnos para poder generar el array

Pega el código de los dos archivos .h en este documento y súbelo junto con el proyecto en zip a la plataforma.

Pon comentarios al código con tus propias palabras.

Se valorará:

El funcionamiento de la primera ventana. max 1,5 ptos (de 5)

El paso entre ventanas max 0,5 pto

El funcionamiento de la segunda ventana max 1,5 ptos

La adecuada explicación en los comentarios max 1,5 ptos

Firma formador/a	
Firma y sello del centro de formación	

```
//Evento del boton "Enviar" para realizar el login
private: System::Void btnEnviar_Click(System::Object^ sender, System::EventArgs^ e) {
    //Los dos campos están definidos con un hint, por lo tanto, antes de obtener el valor, si el valor sigue siendo el hint,
    //se pondrá a vacío
    comprobarFormulario();

    //lectura de los campos de la ventana, se pasa de String^ a string con el #include "msclr/marshal_cppstd.h"
    string password = msclr::interop::marshal_as<std::string>(entradaContrasenia->Text);
    string correo = msclr::interop::marshal_as<std::string>(entradaCorreo->Text);

    //Se comprueba el login, si es correcto, se abrirá la segunda ventana
    //En caso contrario, muestra un mensaje de error
    if (comprobarLogin(correo, password)) {
        //Pasará a la segunda ventana
        Contenido^ Inicio = gcnew Contenido(); //Crear el objeto de la ventana. Necesario #include "Contenido.h"
        this->Visible = false; // Ocultar la ventana actual
        Inicio->ShowDialog(); // Mostrar la segunda ventana pero sigue oculta
        this->Visible = true; // Mostrar la segunda ventana
    }
    else {
        MessageBox::Show("Credenciales incorrectas"); // Mensaje de error
    }

    //Se reinician los campos a sus valores por defecto
    reiniciarFormulario();
}

//Funcion para establecer los campos del login a vacío si no se modifican
private: System::Void comprobarFormulario() {
    if (this->entradaContrasenia->Text->Equals("Introduce tu contraseña: ")) {
        this->entradaContrasenia->Text = "";
    }
    if (this->entradaCorreo->Text->Equals("Introduce tu correo: ")) {
        this->entradaCorreo->Text = "";
    }
}

//Funcion que retorna true o false según si los datos introducidos coinciden con los
private: System::Boolean comprobarLogin(string correo, string password) {
    if (correo == "email@email.com" && password == "password123") {
        return true;
    }
    return false;
}

//Funcion que establece por defecto el hint del formulario
private: System::Void reiniciarFormulario() {
    this->entradaContrasenia->Text = "Introduce tu contraseña: ";
    this->entradaContrasenia->ForeColor = System::Drawing::Color::Silver;
    this->entradaContrasenia->UseSystemPasswordChar = false;
    this->entradaCorreo->Text = "Introduce tu correo: ";
    this->entradaCorreo->ForeColor = System::Drawing::Color::Silver;
}

//Evento del campo de Password para quitar las propiedades del hint al entrar en el
private: System::Void entradaContrasenia_Enter(System::Object^ sender, System::EventArgs^ e) {
    this->entradaContrasenia->Text = "";
    this->entradaContrasenia->ForeColor = System::Drawing::Color::Black;
    this->entradaContrasenia->UseSystemPasswordChar = true;
}

//Evento del campo de Correo para quitar las propiedades del hint al entrar en el
private: System::Void entradaCorreo_Enter(System::Object^ sender, System::EventArgs^ e) {
    this->entradaCorreo->Text = "";
    this->entradaCorreo->ForeColor = System::Drawing::Color::Black;
}
};
```

Contenido.h

```
//Funcion que carga datos en el grid view al cargar la ventana
```

```
private: System::Void Contenido_Load(System::Object^ sender, System::EventArgs^ e) {

    //Array del objeto Alumnos que se inicializa con tres datos
    array<Alumnos^>^ alumnosArray = gcnew array<Alumnos^>
    {
        gcnew Alumnos{ "Juan", 25 },
        gcnew Alumnos{ "María", 30 },
        gcnew Alumnos{ "Carlos", 40 }
    };

    //Funcion for each para establecer cada objeto de Alumnos en una fila del gridView
    int i = 0;
    for each (Alumnos ^ alumno in alumnosArray) {
        dataGridView1->Rows->Add(); //Se añade la fila

        dataGridView1->Rows[i]->Cells[0]->Value = alumno->Nombre; //En la columna nombre se le agrega el valor del
objeto
        dataGridView1->Rows[i]->Cells[1]->Value = alumno->Edad; //En la columna edad se le agrega el valor del objeto
        i++;
    }
}
```

Alumnos.h

```
ref class Alumnos
{
public:
    //Propiedades públicas
    int Edad;
    String^ Nombre;

    //Constructor de la clase con dos parámetros: String^ e int
    Alumnos(String^ fName, int fEdad)
    {
        Nombre = fName;
        Edad = fEdad;
    }
};
```