Ingeniería
Desarrollo de Software

Sesión #1

Lenguajes de programación II

felix.acosta@umi.edu.mx

academia global

UNIVERSIDAD MÉXICO INTERNACIONAL

15-11-2024

TEMARIO

- Objetivo
- Pregunta de sesión
- Ejercicios
- Resumen
- Referencias
- Redes

Ingeniería
Desarrollo de Software

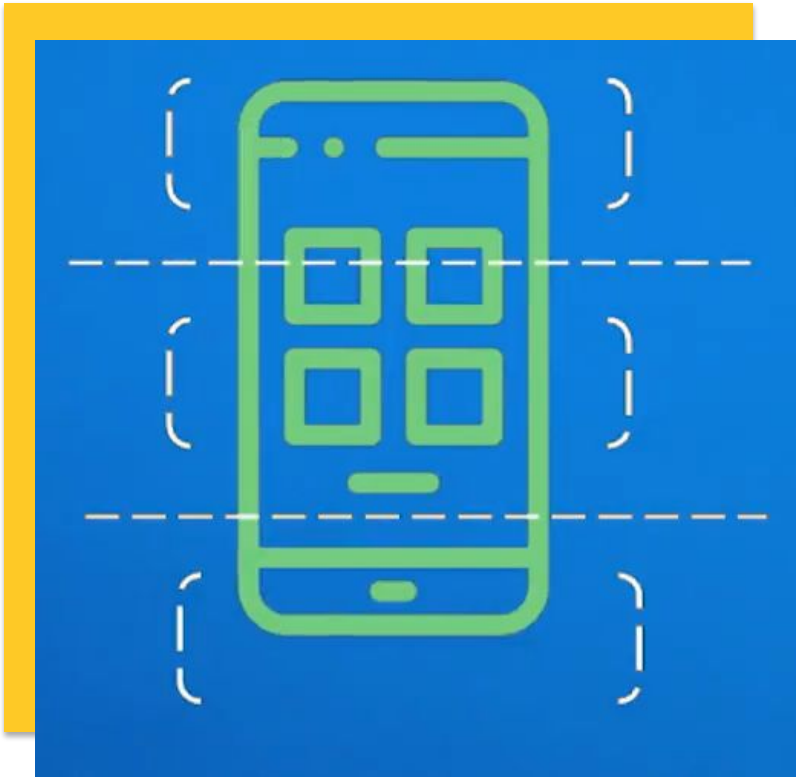
Objetivo

Ingenieria
Desarrollo de Software

Ingenieria
Desarrollo de Software

Hoy aprenderemos a ...

- Utilizar conceptos básicos sobre programación orientada objetos con C++
- Usar el entorno de Visual Studio
- Crear una base de datos con SQL Server



Ingenieria
Desarrollo de Software

Presentar trabajos de preferencia en formato PDF con la siguiente estructura:

- Portada (datos personales)
- Tabla de contenido o índice
- Desarrollo
- Conclusiones
- Bibliografía

Ingeniería
Desarrollo de Software

UNIDAD 1
Herencia
Herencia simple y múltiple

Contestar el Foro 1

Ingenieria
Desarrollo de Software

UNIDAD2

Declaración y atributos locales
Asignación a atributos
Operaciones matemáticas
Llamados a otros métodos
Estructuras de control
Excepciones

Realizar la Actividad#1

Ingenieria
Desarrollo de Software

UNIDAD3
Polimorfismo
Sobrecarga
Sobreescritura

Realizar la Actividad #2

Ingenieria
Desarrollo de Software

UNIDAD 4
Encapsulamiento

Ingeniería
Desarrollo de Software

PROYECTO FINAL

Se necesita una estructura de clases que permita a la empresa UNI controlar los distintos tipos de empleados, así como sus datos personales. Esto se hará a través de clases, herencia de clases y atributos. Las clases, por su parte, deberán ser usadas desde una aplicación donde se gestione la siguiente información:

- Número de Empleado: (autogenerado, numérico)
- Nombre: (capturable, alfanumérico)
- Apellido Paterno: (capturable, alfanumérico)
- Apellido Materno: (capturable, alfanumérico)
- Fecha de Nacimiento: (capturable tipo fecha)
- RFC: (calculado conforme al nombre y fecha de nacimiento, alfanumérico) Centro de Trabajo: (capturable, alfanumérico)
- Puesto: (elegible desde el número de clave con base en el catálogo de puestos)
- Descripción del Puesto: (desplegar según puesto elegido con base en el catálogo)
- Directivo: (bandera para indicar tipo de empleado; para directivo 1; para empleado normal 0)

ACTIVIDAD Con los datos que se ingresaron en la base de datos en la Actividad 1, la conexión a la base de datos y las tablas vacías creadas de la Actividad 2, programar la muestra de datos dentro de las tablas.

Ingenieria
Desarrollo de Software

Programación orientada a objetos

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

¿Qué resuelve la POO?

Ingeniería
Desarrollo de Software

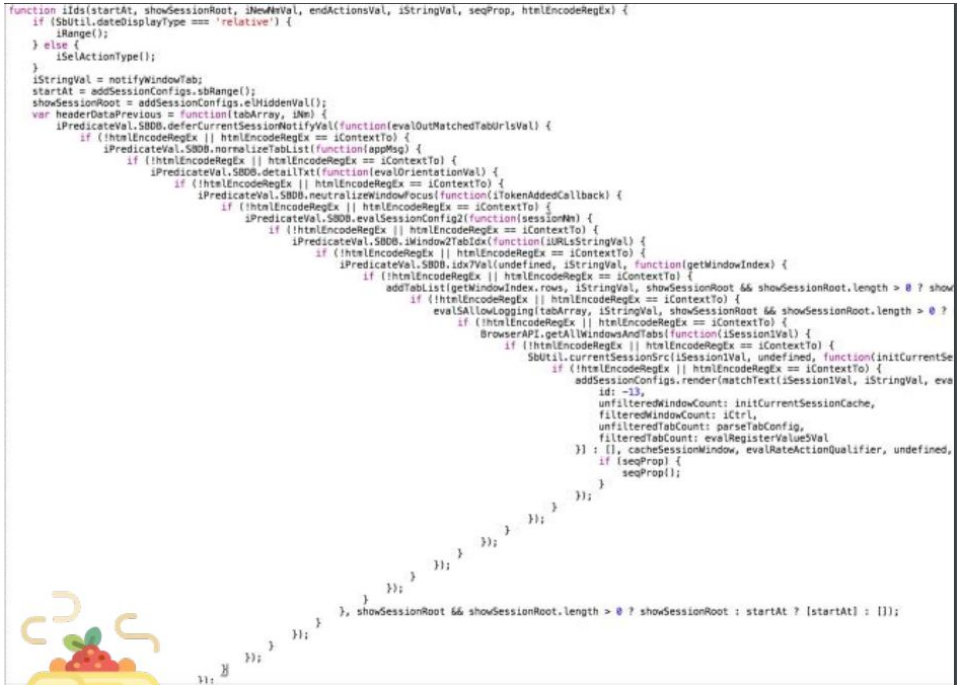
Ingenieria
Desarrollo de Software

¿Qué resuelve?

- Código muy largo
- Si algo falla, todo se rompe
- Difícil de mantener
- Código espagueti

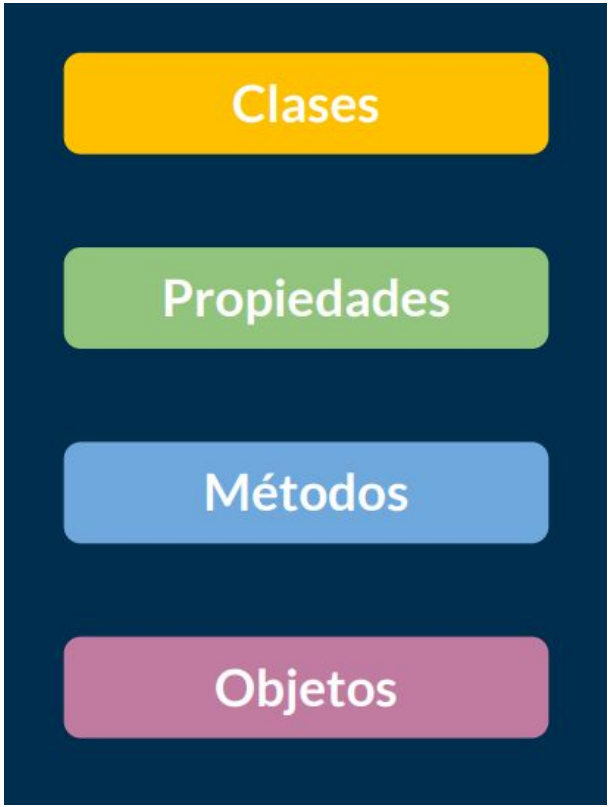
Ingenieria
Desarrollo de Software

Código espagueti



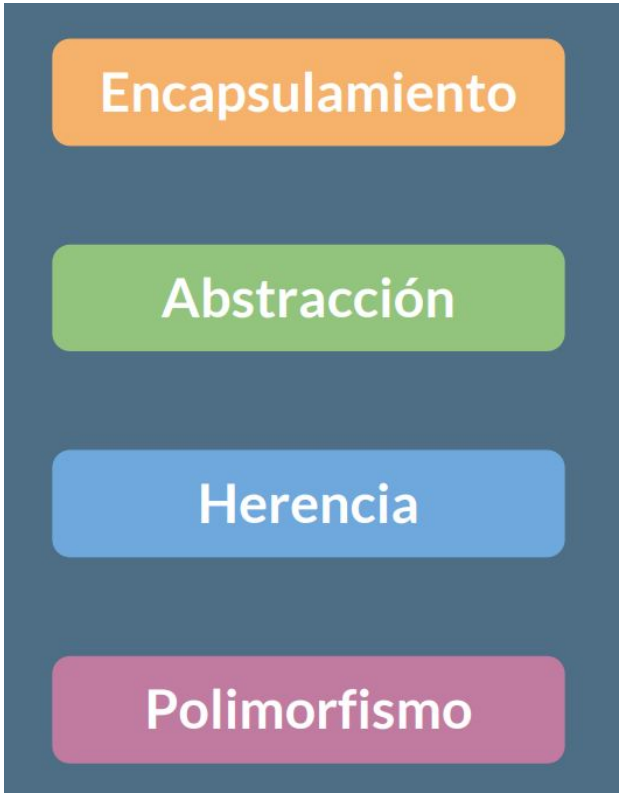
Ingenieria
Desarrollo de Software

Paradigma de POO



Ingenieria
Desarrollo de Software

Pilares de la POO



Ingenieria
Desarrollo de Software

UML (Unified Modeling Language)

- Código muy largo
- Si algo falla, todo se rompe
- Difícil de mantener

Ingenieria
Desarrollo de Software

UML (Unified Modeling Language)

- Clases
- Casos de Uso
- Objetos
- Actividades
- Iteración
- Estados
- Implementación

Ingenieria
Desarrollo de Software

Objetos

Cuando tengamos un problema lo primero que debemos hacer es identificar Objetos

Objetos son aquellos que tienen propiedades y comportamientos

Pueden ser físicos o conceptuales

Ingenieria
Desarrollo de Software

Propiedades

Pueden llamarse atributos serán sustantivos

Nombre, Tamaño, Forma, Estado

Ingenieria
Desarrollo de Software

Comportamientos

Serán todas las operaciones del objeto, suelen ser verbos o sustantivo y verbo

Login()
LogOut()
makeReport()

Ingenieria
Desarrollo de Software

Perro - Propiedades

- Nombre
- Color
- Raza
- Altura

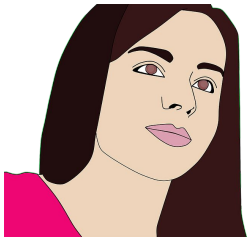
Ingenieria
Desarrollo de Software

Perro - Comportamiento

- Ladrar()
- Comer()
- Dormir()
- Correr()

Ingenieria
Desarrollo de Software

Adopciones

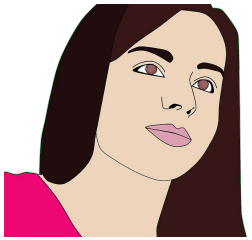


- id
- Nombre
- Color
- Raza
- Altura

- serAdoptado()
- Ladrar()
- Comer()
- Dormir()
- Correr()

Ingenieria
Desarrollo de Software

Adopciones



- id
- Nombre
- Color
- Raza
- Altura

- 001
- Berlín
- Negro con blanco
- Husky
- 0.5 m

Ejercicios

Ingeniería
Desarrollo de Software

Ingenieria
Desarrollo de Software

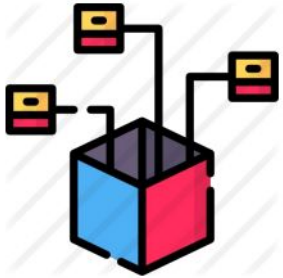
Discusión en grupo (5 minutos)

Ejemplos de “Objetos” con propiedades y comportamientos

Ingenieria
Desarrollo de Software

Clase

- Es el modelo sobre el cual se construirá nuestro objeto
- Las clases me permitirán generar más objetos



Ingenieria
Desarrollo de Software

Clase - Modularidad

- Divide el programa en diferentes partes o módulos / clases
- Separar las clases en archivos

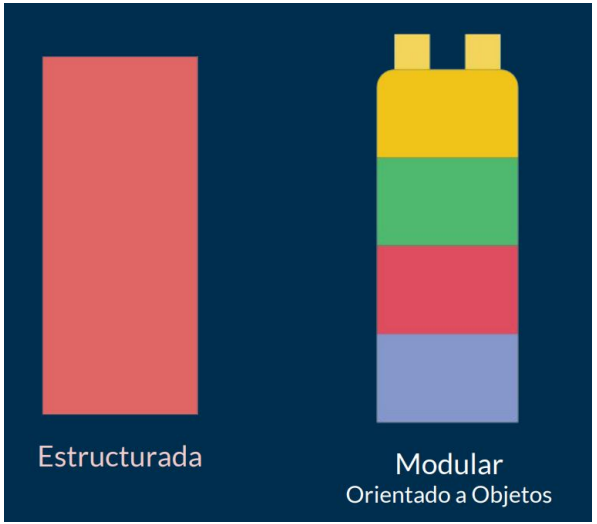
2 Seats + 4 Sides

4 seats + 5 Sides

Ingenieria
Desarrollo de Software

Clase - Modularidad

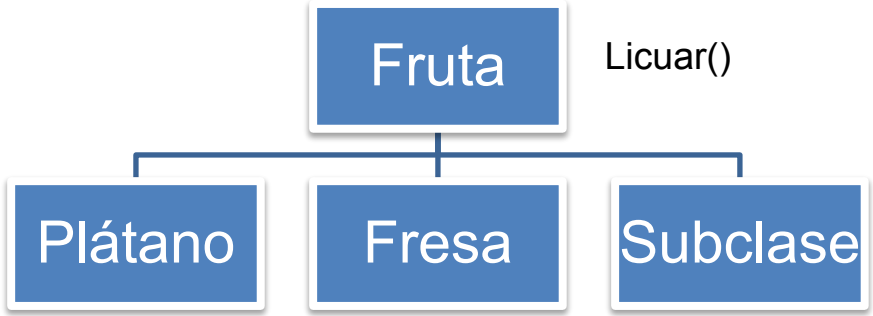
- Reutilizar
- Evitar colapsos
- Mantenible
- Legibilidad
- Resolución rápida de problemas



Ingenieria
Desarrollo de Software

Herencia

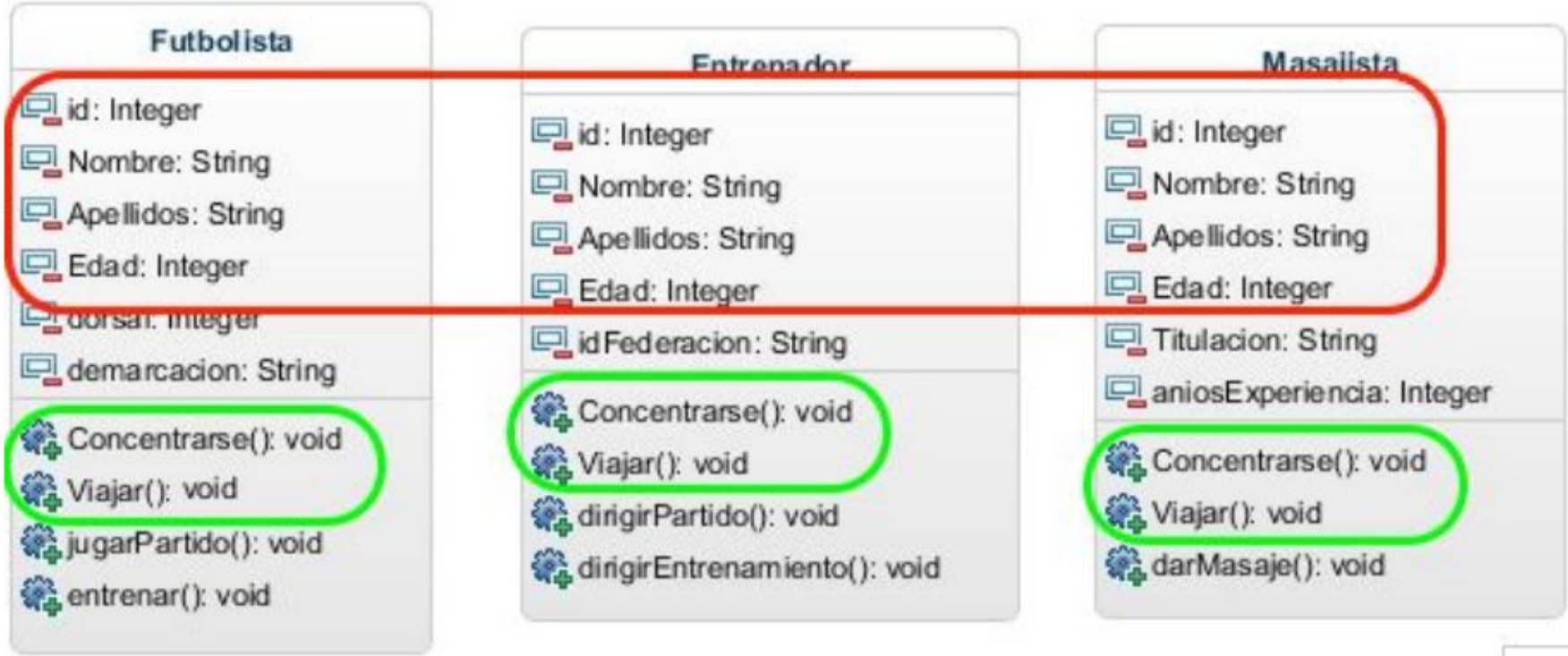
- Crearemos nuevas clases a partir de otras
- Se establece una relación padre-hijo



Color : String
Peso : Float
Estado : String
Licuar()

Tipo: String
Exportar()
Cosechar()

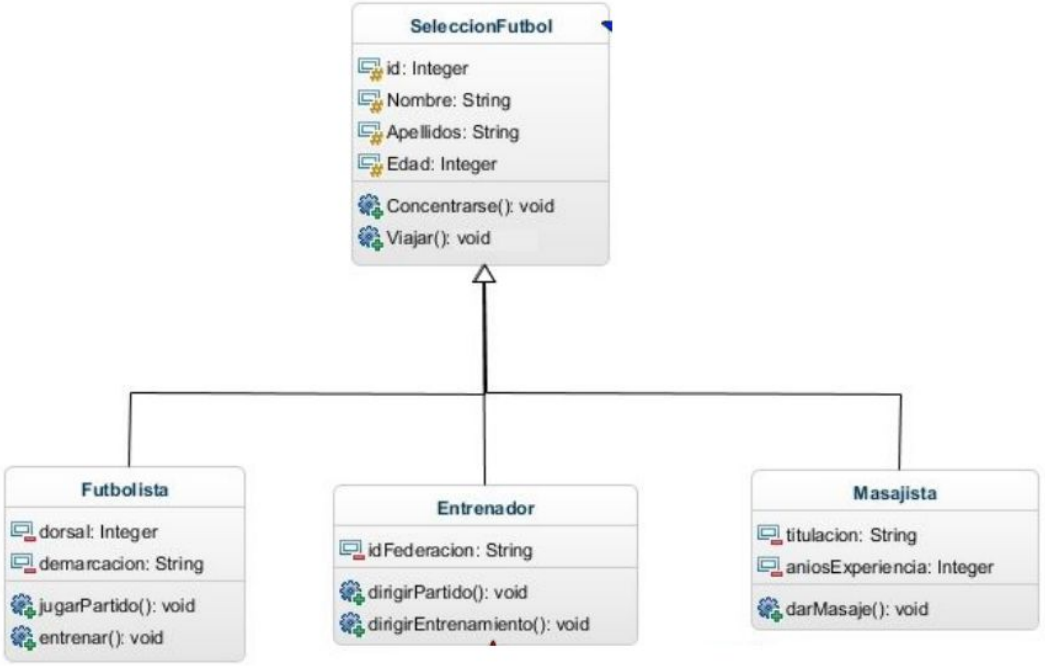
Ingenieria
Desarrollo de Software



Ingenieria
Desarrollo de Software

Herencia

Superclase
SeleccionFutbol



Ingenieria
Desarrollo de Software

Constructor

- Dar un estado inicial al objeto
- Tiene el mismo nombre de la clase
- Son los parámetros mínimos que necesita el objeto para que pueda vivir

Resumen

Ingeniería
Desarrollo de Software

Ingenieria
Desarrollo de Software

Hoy aprendimos a ...

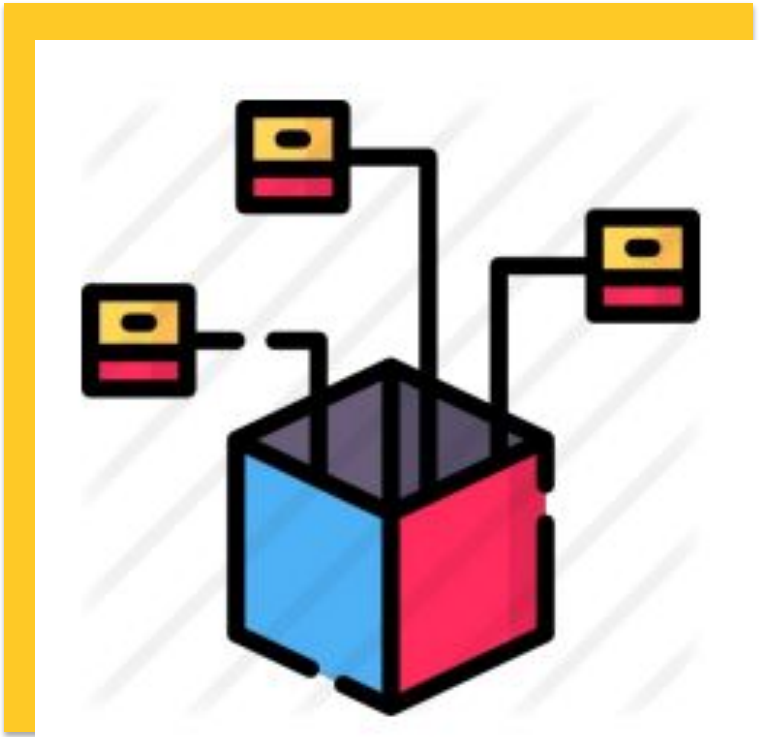
- Reconocer los elementos fundamentales de la POO en C++
- Realizar Actividad #1
- Comentar en el Foro

Referencias

Ingeniería
Desarrollo de Software

Ingenieria
Desarrollo de Software

BRAUNSTEIN, Silvia; L. GIOIA y Alicia B.: Introducción a la programación y a las estructuras de datos. Eudeba, Buenos Aires, 1986



Redes

Ingenieria
Desarrollo de Software

Ingenieria
Desarrollo de Software

Anahí Salgado



Grupos profesionales

Geeks for Geeks

academia global

Ingeniería
Desarrollo de Software

UNIVERSIDAD MÉXICO INTERNACIONAL

Ingeniería
Desarrollo de Software

Sesión #2

Lenguajes de programación II

felix.acosta@umi.edu.mx

academia global

UNIVERSIDAD MÉXICO INTERNACIONAL

19-07-2024

TEMARIO

- Objetivo
- Pregunta de sesión
- Ejercicios
- Resumen
- Referencias
- Redes

Ingeniería
Desarrollo de Software

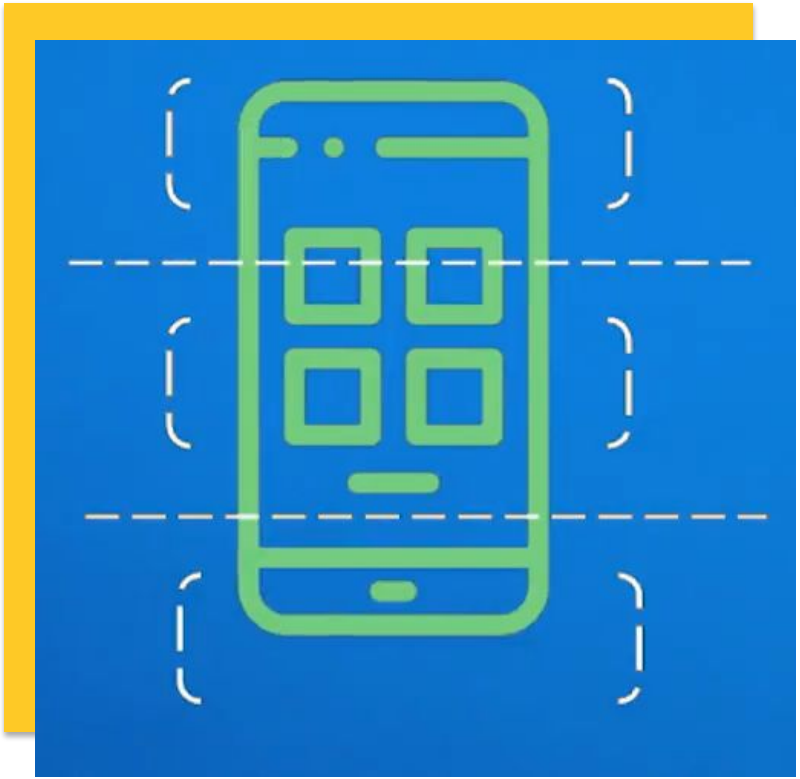
Objetivo

Ingenieria
Desarrollo de Software

Ingenieria
Desarrollo de Software

Hoy aprenderemos a ...

- Utilizar conceptos básicos sobre programación orientada objetos con C++
- Instalar y configurar ODBC
- Conectar con una base de datos con SQL Server in C++
- Consultar información desde una base de datos en SQL Server



Ingenieria
Desarrollo de Software

Descargar ODBC

- Link de descarga

Configurar ODBC

- Buscar ODBC Data Source
- Ir a la pestaña de System DSN
- Agregar un nuevo DSN
- Asignar un nombre (sqlserver)
- En server poner el nombre del SQLServer.
- Elegir la autenticación de windows y conectarse mediante SQLServer
- Elegir la base de datos por default.
- NEXT y FINISH
- Test Data Source

Ingenieria
Desarrollo de Software

Visual Studio Community 2022

- Crear un nuevo proyecto (Console App - C++)
- Si es necesario descargar el C++

```cpp
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <iostream>

using namespace std;

int main() {
    SQLHENV hEnv;
    SQLHDBC hDbc;
    SQLRETURN ret;

    // Asignar un gestor de entorno
    ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
    ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION, (SQLPOINTER)SQL_OV_ODBC3, 0);

    // Asignar un gestor de conexión
    ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);

    // Conectarse a la base de datos
    ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver", SQL_NTS, (SQLWCHAR*)L"Username", SQL_NTS, (SQLWCHAR*)L"Password", SQL_NTS);

    if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
        cout << "Conectado a la base de datos exitosamente." << endl;

    }
    else {
        cout << "Fallo la conexion a la base de datos" << endl;
    }

    // Desconectar y liberar manejadores
    SQLDisconnect(hDbc);
    SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
    SQLFreeHandle(SQL_HANDLE_ENV, hEnv);

    return 0;
}
```

Ingenieria
Desarrollo de Software

```
// Ejemplo de ejecución de una consulta
SQLHSTMT hStmt;
ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);

// Ejemplo de consulta SELECT
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Datos_Empleados", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    SQLCHAR name[255];
    SQLCHAR last_name[255];
    int ID;
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &ID, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_CHAR, name, sizeof(name), NULL);
        SQLGetData(hStmt, 3, SQL_C_CHAR, last_name, sizeof(name), NULL);
       // SQLGetData(hStmt, 2, SQL_C_LONG, &age, 0, NULL);
        cout << "ID: " << ID << ", Name: " << name << ", Lastname: " << last_name << endl;
    }
}

// Liberar el manejador de conexion
SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
```

Ingenieria
Desarrollo de Software

Repositorio de Github

- Conexión a la base de datos

Ingeniería
Desarrollo de Software

Sesión #3

Lenguajes de programación II

felix.acosta@umi.edu.mx

academia global

UNIVERSIDAD MÉXICO INTERNACIONAL

2-08-2024

TEMARIO

- Objetivo
- Pregunta de sesión
- Ejercicios
- Resumen
- Referencias
- Redes

Ingeniería
Desarrollo de Software

Objetivo

Ingenieria
Desarrollo de Software

Ingenieria
Desarrollo de Software

Alinear las tablas

Para mantener los datos alineados cuando los imprimes en la consola, puedes utilizar la libreria `<iomanip>`. Esta biblioteca proporciona funciones como `setw`, que te permiten establecer el ancho de los campos impresos, asegurando que cada columna tenga el mismo ancho y que los datos se alineen correctamente.

Ingenieria
Desarrollo de Software

```
// Imprimir encabezados de la tabla
cout << "+---------------+--------------------+----------------------+" << endl;
cout << "| No. Empleado  | Nombre             | Apellido Paterno     |" << endl;
cout << "+---------------+--------------------+----------------------+" << endl;
```

```
// Ejemplo de consulta SELECT
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Datos_Empleados", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    SQLCHAR name[50];
    SQLCHAR last_name[50];
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_CHAR, name, sizeof(name), NULL);
        SQLGetData(hStmt, 3, SQL_C_CHAR, last_name, sizeof(last_name), NULL);

        // Imprimir datos de la fila con alineación
        cout << "| " << setw(14) << left << num_empleado
            << "| " << setw(18) << left << name
            << "| " << setw(19) << left << last_name << " |" << endl;
    }
}

// Imprimir línea final de la tabla
cout << "+---------------+--------------------+----------------------+" << endl;

// Liberar el manejador de conexión
SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
```

Ingenieria
Desarrollo de Software

Explicación

Librería <iomanip>: Se incluye esta librería para usar manipuladores de flujo como setw y left.

Encabezados y Bordes de la Tabla: Se imprimen las líneas de los bordes superiores e inferiores de la tabla, así como los encabezados de las columnas, utilizando + y -.

Alineación de Datos: Se utiliza **setw** para establecer un ancho fijo para cada columna. **left** se usa para alinear el texto a la izquierda. Esto asegura que cada dato se alinee correctamente dentro de su columna, independientemente de su longitud.

Impresión de Filas: Se imprimen los datos de cada fila, utilizando los mismos anchos de columna especificados para los encabezados, lo que garantiza que los datos se alineen correctamente.