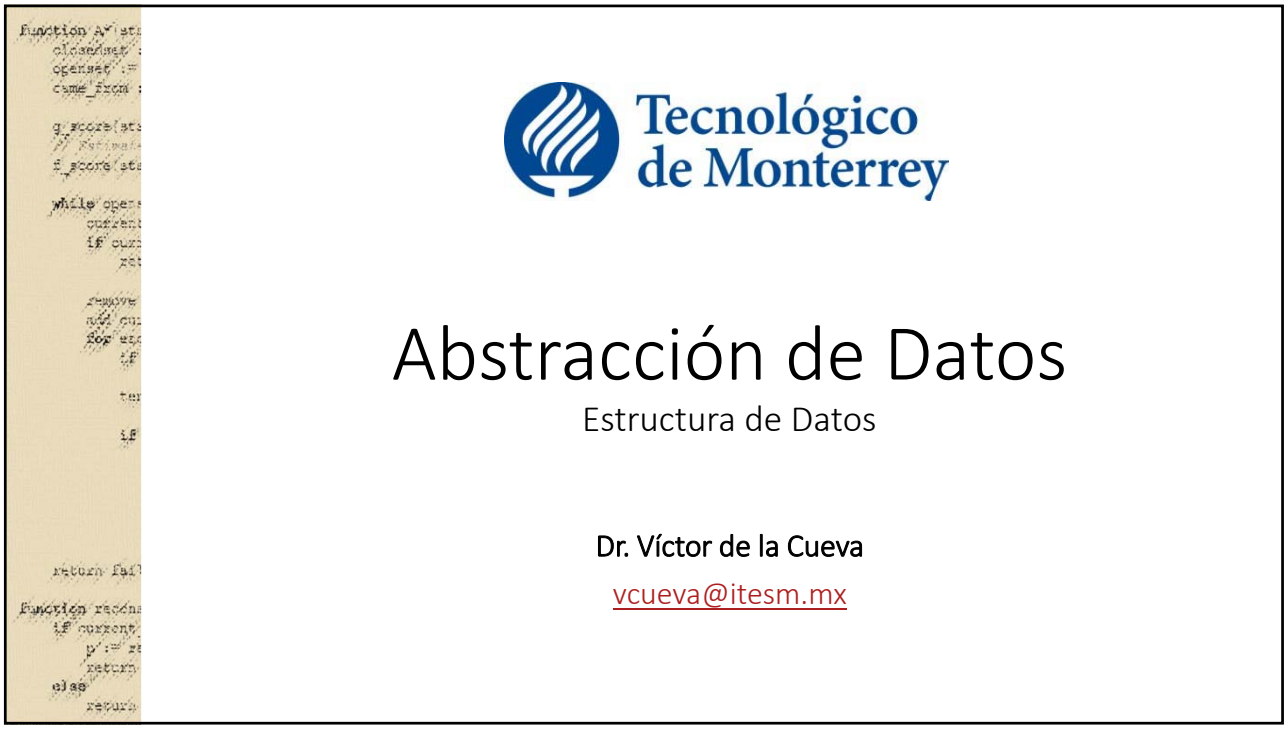


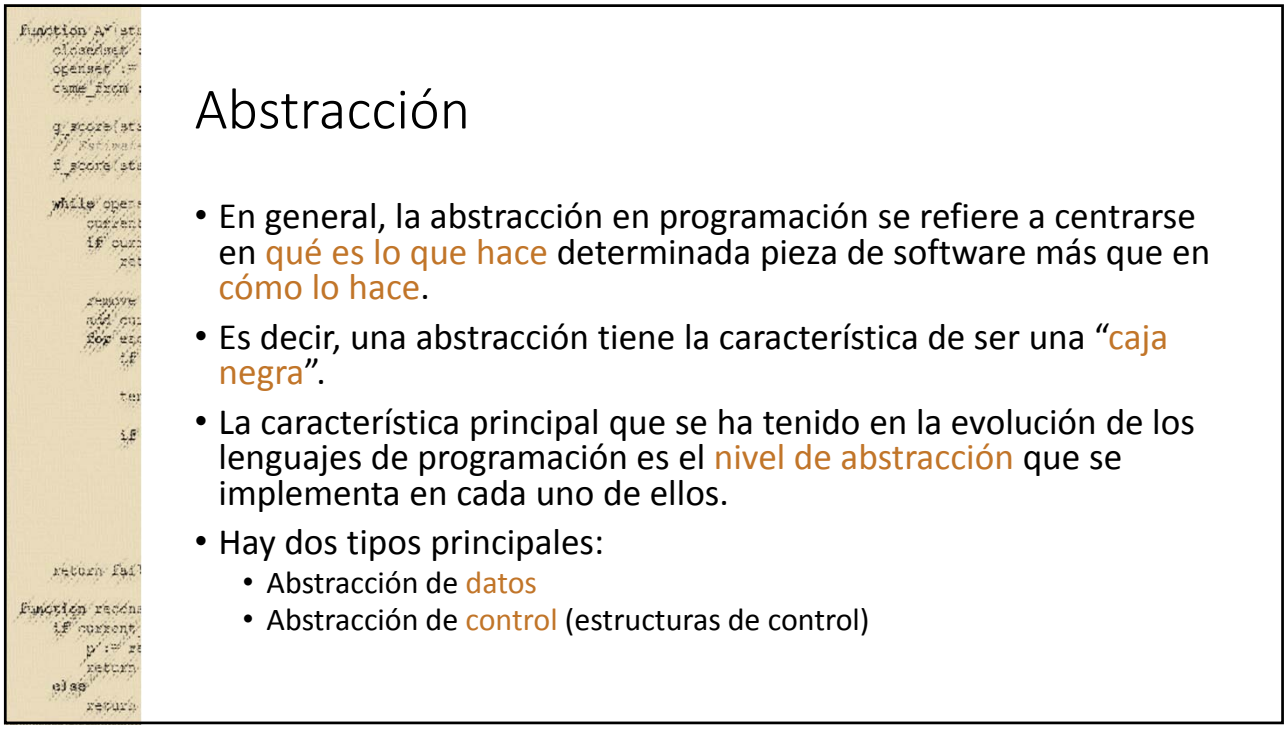
Tecnológico de Monterrey

# Abstracción de Datos

Estructura de Datos

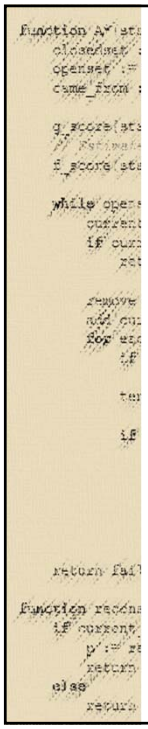
Dr. Víctor de la Cueva

vcueva@itesm.mx



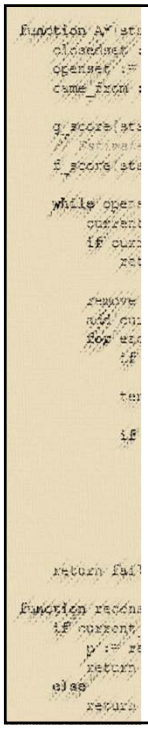
## Abstracción

- En general, la abstracción en programación se refiere a centrarse en qué es lo que hace determinada pieza de software más que en cómo lo hace.
- Es decir, una abstracción tiene la característica de ser una "caja negra".
- La característica principal que se ha tenido en la evolución de los lenguajes de programación es el nivel de abstracción que se implementa en cada uno de ellos.
- Hay dos tipos principales:
  - Abstracción de datos
  - Abstracción de control (estructuras de control)



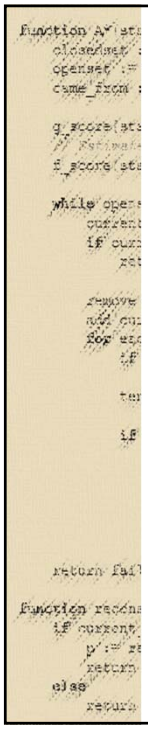
# Ejemplos de abstracción

- Suma de dos números:
  - Se hacía en lenguaje ensamblador y consistía en:
    - Mover un número de la memoria en un registro
    - Mover otro número de la memoria a otro registro
    - Sumar los dos registros y dejar el resultado en uno de ellos
    - Pasar el resultado a la memoria
  - C = A + B
- Un ciclo FOR:
  - Se hacía en lenguaje ensamblador:
    - Colocar un registro a 0 (contador)
    - Verificar su se llegó al número deseado. De ser verdadero, salir del ciclo
    - Si no, incrementar el contador y saltar de regreso a una dirección de memoria para ejecutar nuevamente el cuerpo de instrucciones.
  - FOR i = 0 to N do {…}



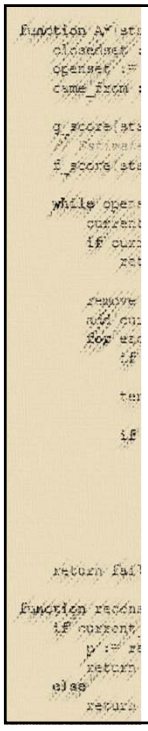
# Abstracción de datos

- La abstracción de datos en un concepto básico en la solución de un problema.
- Permite definir:
  - El dominio y la estructura de los datos
  - El conjunto de atributos que caracterizan a esos datos
  - Las operaciones válidas aplicables sobre dichos datos
- Es decir, la abstracción es el mecanismo por medio del cual se define un concepto general a partir del conocimiento que se tenga de objetos particulares [2].



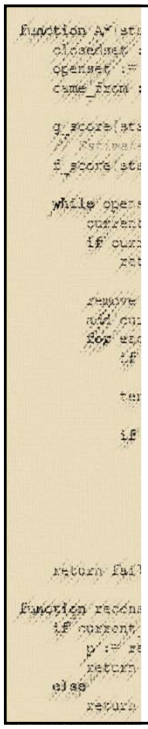
# Tipos de datos abstractos

- Una de nuestras metas es escribir código que pueda ser reusado en muchas diferentes aplicaciones.
- Una forma de lograrlo es encapsular los datos con los métodos que operan sobre estos datos.
- Un nuevo programa puede entonces usar éstos métodos para manipular dichos datos sin tenerse que enterar de los detalles sobre la representación de los datos o la implementación de los algoritmos en los métodos.
- Los datos encapsulados junto con sus métodos es llamado un tipo de dato abstracto (ADT por sus siglas en inglés).



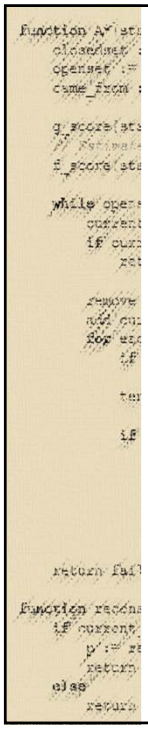
# ADT

- La abstracción da origen a lo que se conoce como un ADT.
- Es un tipo de datos definido por el usuario, cuyas operaciones especifican cómo un cliente (usuario) puede manipular los datos.
- Un ADT constituye un modelo abstracto que define una interfaz entre el usuario y el dato.
- El ADT es independiente de la implementación, lo que permite al diseñador de la solución enfocarse en el modelo de datos y en sus operaciones, sin considerar un lenguaje de programación en particular (luego lo traduce al lenguaje elegido) [2].



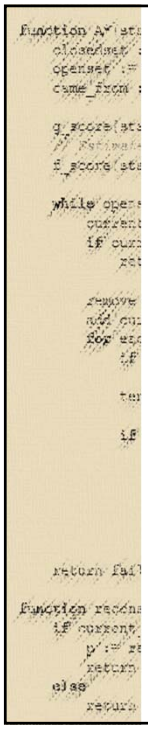
# API

- Los datos encapsulados para una cierta aplicación forman una serie de "instrucciones nuevas" que se ponen a disposición de los usuarios en una *Application Programming Interface* (API).
- Así, existen APIs para:
  - Lenguajes de programación (e.g. API de C++ o de Java)
  - Controlar ciertos dispositivos (e.g. API de drones Phantom),
  - Comunicarse con servidores (e.g. API de Wolfram), etc.
- Cualquiera puede crear una API con su proyecto y si es muy útil y aporta cosas novedosas o mejores a las existentes, su uso se extiende y los lenguajes de programación la pueden incluir como parte de sus librerías.



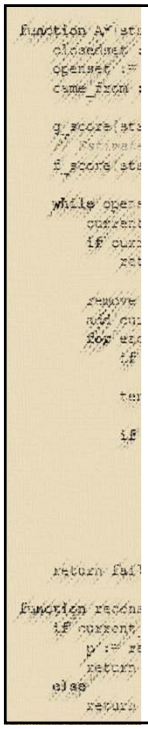
# Objetivo principal del curso

- El objetivo principal del curso es aprender sobre los ADTs que son usados para estructurar datos con el propósito de proporcionar facilidad y eficiencia de:
  - Almacenamiento
  - Organización
  - Procesamiento de información
- Estos ADTs son comúnmente llamados Estructuras de Datos.



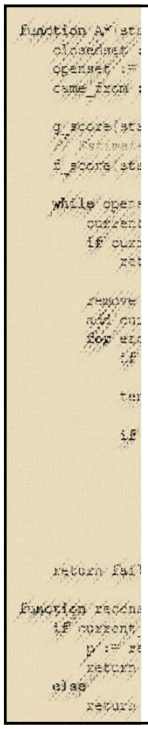
# Ejemplo: ADT de un modelo simplificado [2]

- Dominio: alumnos universitario
- Datos:
  - Nombre: cadena de caracteres
  - Dirección: cadena de caracteres
  - Matrícula: número entero
  - Año de ingreso: número entero
  - Carrera: cadena de caracteres
  - Promedio: número real
- Operaciones válidas definidas para el ADT: representan aquellas operaciones que se pueden realizar sobre o con los datos de un alumno universitario:
  - Actualizar Dirección
  - Actualizar Promedio
  - Actualizar carrera
  - …



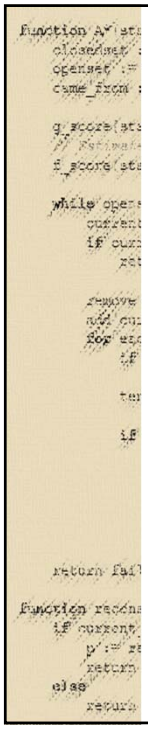
# General vs particular

- El ADT anterior representa a los alumnos universitarios en general (se está describiendo un concepto general).
- Una instancia representa un alumno en particular (con nombre, dirección, carrera, etc.).
- En programación orientada a objetos:
  - ADT es una clase
  - Instancia es un objeto



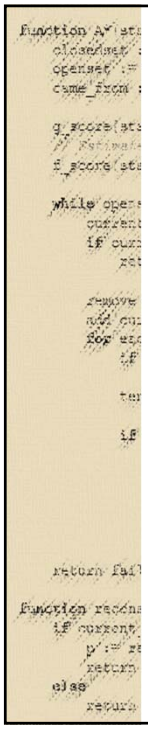
# Clases

- Una clase proporciona una forma de implementar un ADT en C++.
- Si los campos de datos son privados estos puede ser “accesados” solamente a través de métodos públicos.
- Los métodos controlan el acceso a los datos y determinan la forma en la cual los datos son manipulados.



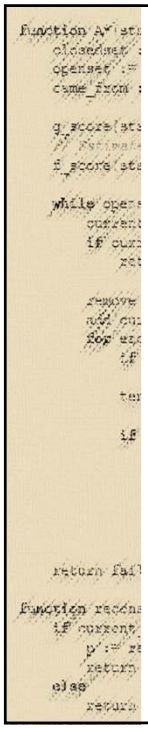
# Elementos de un ADT

- Encabezado: nombre del ADT
- Descripción de los datos:
  - Se especifican los datos y las estructuras correspondientes para representarlos.
  - Los datos constituyen los atributos del concepto u objeto definido por medio del ADT.
- Lista de operaciones: es el conjunto de operaciones que se definen como válidas para el ADT y para cada operación deberá indicarse:
  - Entrada: generalmente, proporcionada por el usuario
  - Precondiciones: establecen la situación en la cual se aplicará la operación
  - Proceso: es el conjunto de acciones que definen la operación
  - Salida: valores regresados al usuario luego de la operación
  - Poscondiciones: las condiciones que deberán cumplirse una vez ejecutada la operación.



# Inicializador

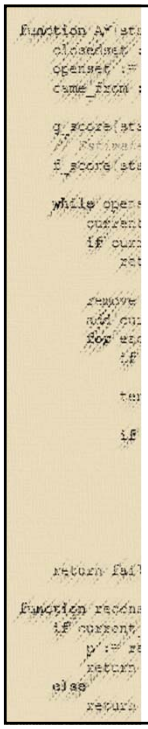
- La mayoría de los ADT tiene una operación especial llamada inicializador, que asigna valores iniciales a los datos.
- Cuando el ADT se implementa por medio de una clase en un lenguaje de programación, esta operación recibe un nombre según el lenguaje empleado.
- En el lenguaje C++ y en Java, el inicializador se conoce como constructor.
- Al memento de declarar un objeto, esta operación lo crea e inicializa.



# Proyecto 1: definición de un ADT

- Definición de la clase Punto, definido por sus coordenadas x e y, con la siguiente API:

```
class Punto {
        private:
                float CoordenadaX, CoordenadaY; // coordenada x y coordenada y del punto
        public:
                Punto(float, float); //inicializa las coordenadas del punto (x,y)
                float ObtenerCoordenadaX(); // regresa la coordenada x del punto
                float ObtenerCoordenadaY(); // regresa la coordenada y del punto
                void ModificaX(float); // cambia el valor de la coordenada x del punto
                void ModificaY(float); // cambia el valor de la coordenada y del punto
                void ImprimeCoordenadas(); // imprime las coordenadas del punto
};
```



# Referencias

[1] Elliot B. Koffman. Objects, abstraction, data structures, and design: using C++. John Wiley & Sons Inc. (2005).

[2] Silvia Guardati. Estructura de datos orientada a objetos: Algoritmos con C++. Pearson (2007).