

The background is a dark blue gradient. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the bottom-left corner, there is a circular inset showing a detailed, grayscale image of a circuit board. In the top-right corner, there is a faint, grayscale image of a circuit board with many small components.

Sintaxis y Semántica del Lenguaje

Docentes y Ayudantes

Comisiones S21 y S22 - Parte práctica

Profesora:

- Jésica Guzmán: jesica.v.guzman@gmail.com

Ayudantes:

- Demian Bogado: lucasbogado@alu.frlp.utn.edu.ar
- Francisco Montiron: franciscomontiron@gmail.com
- Máximo Preneste: maxipreneste@gmail.com
- Juan Casas Ortiz de Rosas: jcasasortizderosas@alu.frlp.utn.edu.ar
- Aylen Tastaca Lugo: aylentastaca@gmail.com
- Francisco Lucich: flucich@alu.frlp.utn.edu.ar
- Nicolás Devecchi: devecchi.ns@gmail.com
- Ariana Wacelinka: arianawacelinka@alu.frlp.utn.edu.ar
- Yael Pilar Luque: ypilarluque@alu.frlp.utn.edu.ar



Temas a desarrollar en la materia

- 01 Tipos Abstractos de Datos
- 02 Autómatas finitos: Determinísticos y No Determinísticos.
- 03 Autómatas de Pila
- 04 Gramáticas de Lenguajes



Aclaraciones para la cursada

- 01 No tienen por qué ser expertos en nada.
- 02 Si está en amarillo, se lo tatúan.
- 03 Sabemos lo que es ser un estudiante (Incluso lo seguimos siendo).
- 04 Por favor no pregunten por las fechas (ni las notas) de los parciales.



Fechas de Parciales (Tentativas)

1º Fecha: 19 de Junio (19/06).

2º Fecha: 3 de Julio (03/07).

3º Fecha: 12 de Julio (12/07).

Fecha Flotante: 31 de Julio (31/07).

Código Classroom: w7f4q6b



TAD Simple: Especificación y Aplicación



Introducción...

Entero, float, caracter, son representaciones de datos simples (números y letras), estos tipos de datos son conocidos como **tipos de datos primitivos**. Pero cuando queremos representar elementos más complejos del problema, necesitamos de una combinación de estos datos.

Por ejemplo: un estudiante tiene un promedio, que podemos representar con una variable de tipo float, nombre/s y apellido/s representables por medio de variables tipo caracter o un legajo que podríamos reflejar con una variable de tipo entero.

Tipos Abstractos de Datos (Definición)

Es entonces cuando surgen los **Tipos Abstractos de Datos (en adelante TAD)**.

Este tipo de datos es definido por el usuario. Cuenta con un conjunto de datos utilizados al definir la **estructura interna** y responde a determinado **comportamiento**.



¿Cuánto mide Alexis?

¿Qué edad tiene?

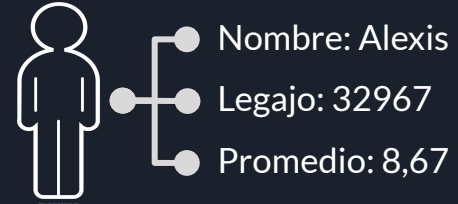
¿Cuál es su color de pelo? ¿Y de ojos?

¿Cuántos hermanos tiene?

Tipos Abstractos de Datos (Características)

Por más que todas las personas cuentan con estos datos, lo que nosotros conocemos no es suficiente para responder estas preguntas. Esto es lo que se conoce como principio de *abstracción*.

Este nuevo tipo de datos trae consigo un nuevo enfoque para la resolución de problemas.





Nuevo enfoque

1

Definición del problema: analizar qué debemos resolver

Identificación de elementos: en esta etapa analizamos los datos pertinentes de las entidades que intervienen en el problema, así como las operaciones a las que deben responder.

2

3

Desarrollo del TAD: una vez identificados los elementos comenzamos a desarrollar la **estructura interna** del TAD, su **especificación** e **implementación** (ésta última propiedad la vemos la clase que viene).

Finalmente pasamos al desarrollo de la aplicación utilizando los TAD creados (*Si ya tuviéramos los TAD podríamos saltar el paso 3 e ir directo al desarrollo de la aplicación*).

4



Estructura Interna

#TADEstudiente

#est=[Nombre, Apellido, Legajo, Promedio]

#est=["", "", 0, 0]

La estructura interna representa cómo está almacenada la información, en este caso una lista. Es muy importante que lo que se pone en cada espacio sea del tipo de dato a utilizar



Especificación

TADEstudiente

#def crearEstudiante():

#crea un estudiante vacío.

#def cargarEstudiante(est, nom, ape, leg, prom):

#Inicia el estudiante con sus datos.

#def asignarEstudiante(est1, est2):

#Asigna los datos de un libro en otro (copia el libro).



Especificación → Ver y Modificar

Al desarrollar TAD, se deben generar métodos que permitan ver y modificar cada atributo del mismo por separado.

#verNombre(estudiante)

#Retorna el nombre del estudiante

#verApellido(estudiante)

#Retorna el apellido del estudiante

#verLegajo(estudiante)

#Retorna el legajo del estudiante

#verPromedio(estudiante)

#Retorna el promedio del estudiante

#modNombre(estudiante, nuevoNom)

#Modifica el nombre del estudiante

#modApellido(estudiante, nuevoApe)

#Modifica el apellido del estudiante

#modLegajo(estudiante, nuevoLeg)

#Modifica el legajo del estudiante

#modPromedio(estudiante, nuevoProm)

#Modifica el promedio del estudiante



Tipos Abstractos de Datos (Características)

Si prestamos atención, en la especificación del TAD no podemos ver cómo funciona cada método definido. Esto se debe a que hay otro principio al que responden los TAD, el de **ocultación**.

Si bien conocemos los datos de Alexis, no tenemos idea cómo están almacenados internamente, y tampoco conocemos cómo se implementan sus métodos, sólo conocemos para qué sirven.



Métodos de un TAD Simple

Los **TAD Simple** representan una entidad específica con su estructura interna y responden a determinados métodos. A lo largo de la materia, trabajaremos siempre con los mismos métodos:

- crear()
- cargar()
- ver()
- modificar()
- asignar()/copiar() **son el mismo método**

Entonces... ¿Puedo crear métodos para todo?



Ejercicio

- Crear dos estudiantes y cargarlos
- Imprimir los datos de cada estudiante al finalizar su respectiva carga
- Comparar los promedios e imprimir un cartel que diga que estudiante tiene mayor promedio.