**Primer Proyecto Programado.**

Marvin Jesus Calvo Acuña

mcalvoa@ucenfotec.ac.cr

**RESUMEN:***.*

**PALABRAS CLAVE**: Lista, Nodo, ListaPersona, ListaCurso, ListaGeneralEstudiante.

# INTRODUCCIÓN

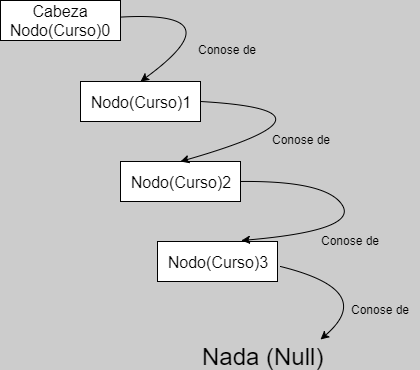
En este documento se describe los tipos de datos abstractos utilizados en la solución del primer proyecto de Estructuras de datos 1 impartido en la Universidad Cenfotec. Se presentará de cada estructura de datos creada la representación abstracta,la invariante del TAD y las operaciones del mismo.

# LISTA CURSO

## REPRESENTACIÓN ABSTRACTA

<Curso1, Curso2, ..., Curso>,

dónde Curso sea diferente de nulo.



Se puede notar en esta representación como cada objeto Nodo referencia al siguiente y así sucesivamente hasta llegar al nodo que referencia a nada o nulo.

## INVARIANTE

Todos los Nodos son elementos de la Lista Curso y contienen un objeto Curso como atributo.

El curso no puede estar repetido

y se registran en la lista siempre al inicio.

## OPERACIONES

ListaCurso(Nada)->ListaCurso (Constructora) //Por defecto devuelve una lista vacía.

**esVacia**(Nada)->Booleano

(Analizadora)//Retorna true si está vacía y fase de lo contrario.

**getCabeza**(Nada)->Nodo

(Analizadora)//Retorna el primer elemento de la lista y hay que tener en cuenta si la lista está vacía.

**setCabeza**(Nodo)->Nada

(Modificadora)//Modifica la cabeza de la lista.

**getTaminio**(Nada)->int

(Analizadora)//Retorna el tamaño de la lista.

**setTamanio**(int)->Nada

(Modificadora)//Modifica el contador del tamaño de lista.

**agregarAlInicio**(Curso)->Nada

(Modificadora)//Agrega un nodo al inicio de la lista.

**buscar**(String)->Curso

(Analizadora)//Busca el Curso según el código y si lo encuentra retorna el curso de lo contrario retorna null.

**listar**(Nada)->nada

(Salida en Pantalla)//Muestra en pantallas el contenido de sus cursos que contiene.

## MANEJO DEL ERROR

Toda las operaciones de la lista deben mantener invariante del TAD.

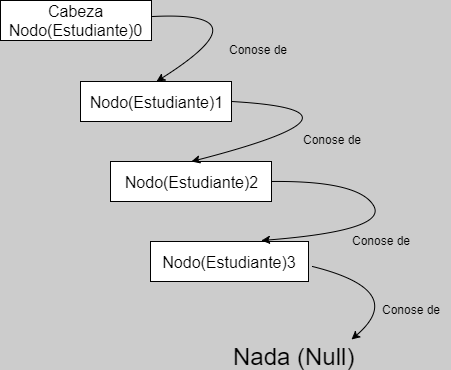
Cuando ocurra una operación indebida en la lista o que genere error la operacion debe abortar su ejecución, y se podría retornar un valor booleano según sea el caso .

# LISTA PERSONA

## REPRESENTACIÓN ABSTRACTA

<Estdiante1, Estdiante2, ..., Estdiante>,

dónde Estudiante sea diferente de nulo.



Se puede notar en esta representación como cada objeto Nodo referencia al siguiente y así sucesivamente hasta llegar al nodo que referencia a nada o nulo.

## INVARIANTE

Todos los Nodos son elementos de la Lista Persona y contienen un objeto Persona como atributo.

La Persona no puede estar repetida

y se registran en la lista siempre al inicio.

## OPERACIONES

**ListaPersona**(Nada)->ListaPersona (Constructora) //Por defecto devuelve una lista vacía.

**esVacia**(Nada)->Booleano

(Analizadora)//Retorna true si está vacía y fase de lo contrario.

**getCabeza**(Nada)->Nodo

(Analizadora)//Retorna el primer elemento de la lista y hay que tener en cuenta si la lista está vacía.

**setCabeza**(Nodo)->Nada

(Modificadora)//Modifica la cabeza de la lista.

**getTaminio**(Nada)->int

(Analizadora)//Retorna el tamaño de la lista.

**setTamanio**(int)->Nada

(Modificadora)//Modifica el contador del tamaño de lista.

**agregarAlInicio**(Persona)->Nada

(Modificadora)//Agrega un nodo al inicio de la lista.

**buscar**(String)->Persona

(Analizadora)//Busca la Persona según lacedula y si lo encuentra retorna el curso de lo contrario retorna null.

**listar**(Nada)->nada

(Salida en Pantalla)//Muestra en pantallas el contenido de las personas .

## MANEJO DEL ERROR

Toda las operaciones de la lista deben mantener invariante del TAD.

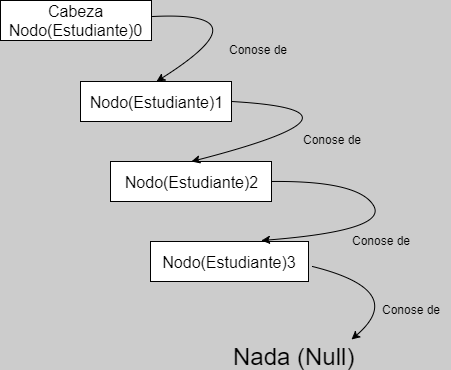
Cuando ocurra una operación indebida en la lista o que genere error la operacion debe abortar su ejecución, y se podría retornar un valor booleano según sea el caso .

# LISTA GENERAL DE ESTUDIANTES

## REPRESENTACIÓN ABSTRACTA

<Estdiante1, Estdiante2, ..., Estdiante>,

dónde Estudiante sea diferente de nulo.



Se puede notar en esta representación como cada objeto Nodo referencia al siguiente y así sucesivamente hasta llegar al nodo que referencia a nada o nulo.

## INVARIANTE

Todos los Nodos son elementos de la Lista Persona y contienen un objeto Persona como atributo.

La Persona no puede estar repetida

y se registran en la lista siempre al inicio.

## OPERACIONES

**ListaGeneralEstudiante**(Nada)->ListaGeneralEstudiante (Constructora) //Por defecto devuelve una lista vacía.

**esVacia**(Nada)->Booleano

(Analizadora)//Retorna true si está vacía y fase de lo contrario.

**getCabeza**(Nada)->Nodo

(Analizadora)//Retorna el primer elemento de la lista y hay que tener en cuenta si la lista está vacía.

**setCabeza**(Nodo)->Nada

(Modificadora)//Modifica la cabeza de la lista.

**getTaminio**(Nada)->int

(Analizadora)//Retorna el tamaño de la lista.

**setTamanio**(int)->Nada

(Modificadora)//Modifica el contador del tamaño de lista.

**agregarAlInicio**(Persona)->Nada

(Modificadora)//Agrega un nodo al inicio de la lista.

**buscar**(String)->Persona

(Analizadora)//Busca la Persona según lacedula y si lo encuentra retorna el curso de lo contrario retorna null.

**listar**(Nada)->nada

(Salida en Pantalla)//Muestra en pantallas el contenido de las personas .

**listarEstudiantePorEstado**(String)->Nada (Salida en Pantalla)//Muestra en pantalla los datos de los estudiante según el estado .

## MANEJO DEL ERROR

Toda las operaciones de la lista deben mantener invariante del TAD.

Cuando ocurra una operación indebida en la lista o que genere error la operacion debe abortar su ejecución, según sea el caso.

# REFERENCIAS

[1] Profe.Maria Eugenia Ucrós,Profe.Christian Sibaja Fernández ,UniversidadCenfotec.Presentacion .Estructuras de Datos TDS1402 y TDS2311.