# http://vectorblog.org/wp-content/uploads/2013/06/Telehealth-cartoon-e1370345782337.jpgHoja de trabajo No. 2: Almacenamiento íntegro

Objetivos de la hoja de trabajo

* Recordar concepto de encapsulamiento.
* Recordar concepto de tipo de dato abstracto.
* Introducir al estudiante al tipo de dato abstracto *Pila*.

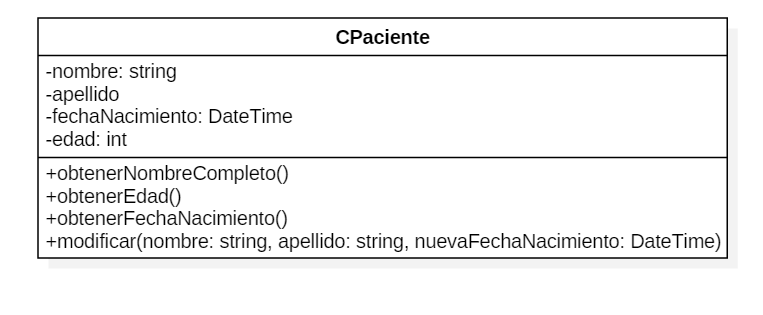
Recomendación: trabajar en grupos de dos integrantes.

## Parte 1: Encapsulamiento y tipo de dato abstracto

En la clase anterior, se planteó el requerimiento de guardar la fecha de nacimiento y calcular la edad del paciente. Suponga que en la clase CPaciente se define de la siguiente manera:

* Nombre de la clase: CPaciente.
* Atributos:
  + Privado string nombre.
  + Privado string apellido.
  + Privado DateTime fechaNacimiento.
  + Privado int edad.
* Constructor:
  + público CPaciente(string unNombre, string unApellido, DateTime laFechaNacimiento)
* Métodos:
  + Público string obtenerNombreCompleto()
  + Público int obtenerEdad()
  + Público DateTime obtenerFechaNacimiento()
  + Público void modificar(string nuevoNombre, string nuevoApellido, DateTime nuevaFechaNacimiento)

1. Creo que hubiera sido más fácil describir esta clase a través de un diagrama de clases de UML. Coloque un diagrama UML de clase que ilustre la anterior definición de la clase Cpaciente.



1. Ahora enfoquémonos en el constructor. ¿Por qué no recibe de parámetro la edad? ¿Cómo calculará la edad? Explique brevemente las razones; **no** es necesario detallar un algoritmo para calcular la edad.

- No recibe de parámetro la edad para evitar errores que el usuario podría cometer en el su calculo. La edad la calcularemos con los métodos ya proveídos por la clase DateTime.

1. Entonces, ¿qué nos promete el constructor? ¿Podríamos decir que nos promete calcular la edad en base a la fecha de nacimiento? ¿Hace algo adicional a calcular la edad?

- el constructor promete asignar el valor correcto a la edad. Si, se calcula la edad basada en la fecha de nacimiento. El constructor también asigna el nombre ya la edad.

A la promesa del constructor, le podemos llamar “pos condición”. Es la condición que tendrá el objeto de tipo CPaciente, después de invocar el constructor. Podríamos describir este método así:

* Constructor: CPaciente
  + **Recibe de entrada**:
    - unNombre de tipo string
    - unApellido de tipo string
    - laFechaNacimiento de tipo DateTime
  + **Devuelve**:
    - Una instancia de CPaciente.
  + **Poscondición**:
    - La instancia creada de CPaciente tendrá asignado el nombre especificado (por parámetro), el apellido especificado, la fecha de nacimiento especificada y adicionalmente tendrá la edad (en años), calculada a partir de la fecha de nacimiento.

1. Pregunta: ¿La anterior definición dice cómo se calcula la edad (detalladamente)?

- no detalladamente, solo sabemos que se calcula a partir de la fecha de edad.

Hay varios detalles que la anterior definición no aclara. Por ejemplo, el algoritmo detallado para calcular la edad. Tampoco especifica si la información se guardará cifrada o si se guardará tal y como la ingresó el usuario. No se dice si se guardarán los datos en un archivo. Pero sí nos dice lo suficiente para que nosotros como programadores podamos usar la clase CPaciente.

1. Complete la siguiente definición del método modificar.

* Método: modificar
  + **Recibe de entrada**:
    - nombre : string
    - apellido : string
    - fechaNacimiento : DateTime
  + **Devuelve**:
    - void
  + **Poscondición**:
    - Se ejecutarán los cambios de nombre, apellido y fecha al paciente.

Si un método no altera ningún atributo (o dato interno) del objeto, decimos que la poscondición es “Los datos permanecen sin cambio”.

1. Complete la siguiente definición del método obtenerNombreCompleto.

* Método: obtenerNombreCompleto
  + **Recibe de entrada**:
    - no recibe entradas.
  + **Devuelve**:
    - la concatenación del nombre y apellido, separados por un carácter vacío.
  + **Poscondición**:
    - Los datos permanecen sin cambio.

***Interacción con otro grupo (I)***: Tome 5 minutos para dialogar con otro grupo sobre sus respuestas a las preguntas 5 y 6.

1. Una última pregunta sobre CPaciente. ¿Qué relación existe entre los datos internos –apellido, nombre, fechaNacimiento, edad–?

Probablemente no podemos decir mucho entre una relación entre apellido y nombre. Pero sí existe una relación entre la fecha de nacimiento y la edad: el último dato se calcula en base al primero.

Recordemos que al definir un tipo de dato abstracto, debemos especificar claramente cómo se relacionan los componentes (o datos internos) y debemos definir claramente las operaciones que se pueden realizar con dichos datos.

Bastaría con definir claramente (entradas, salidas y poscondición) de las operaciones obtenerFechaNacimiento y obtenerEdad para completar la definición de nuestro tipo de dato abstracto CPaciente. Lo cual queda para ejercicio del lector 

En el libro Data Structures and Program Design in C++, se presenta la siguiente definición de *Tipo de dato*:



1. Si **int**, **bool** y **char** son tipos de dato, mencione dos posibles **valores** de cada uno de estos tipos.

* Int: 0,1
* bool: true, false
* char: ‘a’ , ‘b’

1. Si **string** e **int[3]** son tipos de dato, mencione dos posibles **valores** de cada uno de estos tipos.

* String: “string” , “dato”
* {0,0,0} , {1,2,3}

1. ¿Cuál es la diferencia entre los tipos de dato de la pregunta 8 y los tipos de dato de la pregunta 9?

- los de la pregunta 8 son los llamados primitivos, y la pregunta 9 tipo referencia.

1. ¿Qué datos tendrá un objeto de CPaciente, si se invoca al constructor con los siguientes parámetros?  
   CPaciente pacienteEmergencia = new CPaciente(“Karla”, “Quiñonez”, new DateTime(1990, 4, 2));

- nombre: “Karla”

- apellido: “ Quiñonez”

- fechaNacimiento: DateTime(1990,4,2) → year/month/day

- edad: 27

1. Si **CPaciente** es un tipo de dato (dijimos que lo habíamos definido como un tipo de dato abstracto ¿o no?), indique dos posibles **valores** de Cpaciente.

1) – nombre: “juan”

- apellido: “perez”

- fechaNacimiento: DateTime(1997,6,26)

- edad: 20

2) -nombre: “eristalpelgio”

- apellido: “benitez”

- fechaNacimiento: DateTime(1960,5,7)

- edad: 57

***Interacción con otro grupo (II)***: Tome 5 minutos para dialogar con otro grupo sobre sus respuestas a las preguntas cinco preguntas anteriores.

1. En aplicaciones de .NET, ¿ha utilizado la clase List<CPaciente>? Si no la ha utilizado, puede consultar el siguiente enlace:
   1. <https://msdn.microsoft.com/en-us/library/6sh2ey19(v=vs.110).aspx>
2. ¿Cómo definiría la *relación entre los componentes* de un objeto **List**?

- todos los componentes de un list son del mismo tipo

- tienen una secuencia, excepto por el primero y el último, todos tienen un elemento antes y uno después.

1. ¿Cómo definiría las entradas, salidas y poscondición del método **void Add(CPaciente)**?

* Entradas: Objeto de tipo Cpaciente
* Salidas: void
* Poscondición: El objeto Cpaciente es agregado a la lista.

1. Suponga que se ejecutan las siguientes líneas de código. ¿Qué datos tendrá el objeto de tipo List<CPaciente>?  
     
   List<CPaciente> listaMaternidad = new List<CPaciente>();  
   CPaciente primeraMama = new CPaciente(“Nancy”, “Godoy”, new DateTime(1985, 3, 1));  
   listaMaternidad.Add(primeraMama);  
   CPaciente segundaMama = new CPaciente(“Irma”, “Santos”, new DateTime(1986, 1, 15));  
   listaMaternidad.Add(segundaMama);

Mejor respondemos con una imagen :)



1. Indique dos posibles **valores** de **List<CPaciente>**.

* Cpaciente
* Cpaciente

1. ¿Cree que **List<CPaciente>** es un tipo de dato abstracto? ¿Qué se requiere para que sea un tipo de dato abstracto?

- si, la relacion entre sus componentes esta bien definida, al igual que todas las operaciones que podemos realizar con los mismos.

***Interacción con otros grupos (III)****. Haga una pausa y exponga y discuta con otro grupo sus respuestas a las cinco preguntas anteriores.*

## Parte 2: Tipo de dato abstracto *Pila*

Lectura:

* Artículo en Wikipedia: Stack (Abstract Data Type). Disponible en:  
  <https://en.wikipedia.org/wiki/Stack_(abstract_data_type)>
* Referencia técnica de la clase Stack<T> en Microsoft Developer Network. Disponible en:  
  <https://msdn.microsoft.com/en-us/library/3278tedw(v=vs.110).aspx>

1. ¿Cuál es el resultado del siguiente código?  
     
   Stack<int> unaPila = new Stack<int>();  
   unaPila.Push(450);  
   unaPila.Push(198);  
   unaPila.Push(200);  
   int unNumero = unaPila.Pop();  
   MessageBox.Show(unNumero.ToString());  
   int unNumero = unaPila.Pop();  
   MessageBox.Show(unNumero.ToString());  
   int unNumero = unaPila.Pop();  
   MessageBox.Show(unNumero.ToString());

* el primer MessageBox muestra el numero 200
* el segundo MesssageBox muestra el numero 198
* el tercer MessageBox muestra el numero 450

1. Defina las entradas, salidas y poscondición del método Push.

* Entradas: el nuevo dato de tipo T que se desea ingresar al stack.
* Salidas: podria lanzar una Exception si el stack esta lleno (no se si esto cuenta como salida)?
* Poscondicion: (tal ves va por aca) Si el stack no esta lleno, se inserta el dato en el top del stack. Si el stack esta lleno, se lanza una excepcion.

1. Defina las entradas, salidas y poscondición del método Pop.

* Entradas: ninguna
* Salidas: Si el stack esta vacio, se lanza una excepcion. Si no esta vacio, se devuelve el dato que este en el top del stack.
* Poscondicion: Si el stack esta vacio, se lanza una excepcion. Si no esta vacio, el dato que esta en el top del stack se remueve del mismo.

1. Defina las entradas, salidas y poscondición del método Peek. ¿Cuál es la diferencia entre Pop y Peek?

* Entradas: ninguna
* Salidas: Si no esta vacio, se devuelve el dato que esta en el top del stack. Si esta vacio, se lanza una excepcion.
* Poscondicion: Los datos permanecen sin cambio.

1. Si **Stack<int>** es un tipo de dato, de ejemplos de dos valores de **Stack<int>**. De preferencia colocar un dibujo (o diagrama).

* [1,2,3]
* [0,0,0,0,0]

1. ¿Cuál es la relación entre los componentes de una pila?

La pila funciona bajo el principio de Last In, First Out, el unico componente al que se tiene acceso es el que esta en el top del stack.

1. ¿Puede una pila ser un tipo de dato abstracto?

- si

***Interacción con otros grupos (IV)****. Compare sus respuestas anteriores.*

Para entretenerse:

Juego de movilización de vagones en el cual se utiliza un concepto de pila.   
<http://www.coolmath-games.com/0-railroad-shunting-puzzle-2>  
¿De qué manera? ¿En qué varía?

El tren y sus vagones son un stack. Varia en que se pueden quitar o agregar varios vagones con una sola operacion.