
PROYECTO 2 – EDITOR DE IMÁGENES

Daniel Enrique Coti Peñate – Carnet: 201801146

Resumen

Basándose en la compilación de imágenes (específicamente en bits) y conociendo que las imágenes son matrices, se realiza una prueba a un nivel fácil ya que su simplicidad es menor a la real en la ejecución de esta. La cual consiste en que por medio de un archivo XML se podrá representar una o varias imágenes por medio de los signos (*, - ó espacio) y agregándolas en una matriz para realizar dicha imagen, en este caso en específico se menciona que por medio de la teoría de las TDA'S (Tipos de Datos Abstractos) y la teoría de listas en concreto, se realiza una matriz ortogonal con el fin de poder colocar dichos caracteres antes mencionados, y posteriormente graficar dicha imagen en una etiqueta; seguido de ello se podrá realizar distintas operaciones para una o varias imágenes operadas como matrices, entre dichas operaciones podemos mencionar como: la rotación de forma Horizontal y forma Vertical, Unir dos imágenes y encontrar diferencias entre ellas; y a su vez realizar un informe de los cambios que se realizaron.

Summary

Based on the compilation of images (specifically in bits) and knowing that the images are arrays, a test is performed at an easy level since their simplicity is less than the real one in the execution of this. Which is that one or more images can be represented by means of an XML file by means of the signs (*, - or space) and adding them in an array to make that image, in this specific case it is mentioned that through the theory of ADT (Abstract Data Types) and the theory of lists in particular, an orthogonal matrix is made in order to be able to place those aforementioned characters, and then plot that image on a label; following this you can perform different operations for one or more images operated as matrices, among these operations we can mention how: rotation horizontally and vertically, Join two images and find differences between them; and in turn report the changes that were made.

Palabras clave

GUI:

Comúnmente conocido como *Interfaz Grafica de Usuario*, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

MATRIZ:

Las matrices o como algunos las llaman "arreglos multidimensionales" son una estructura de datos bastante similar a los vectores o arreglos. De hecho, una matriz no es más que una serie de vectores contenidos uno en el otro (u otros), es decir, una matriz es un vector cuyas posiciones son otros vectores. Hablemos con más detalle de esto para quedar más claros.

LISTAS:

Una lista es una colección ordenada de valores. Una lista puede contener cualquier cosa. Las listas son estructuras lineales y dinámicas de datos. La principal ventaja del dinamismo lo representa el hecho de que se adquieren posiciones de memoria a medida que se necesitan y se liberan cuando ya no se requieren.

PILA:

Es una lista ordenada o estructura de datos que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés Last In, First Out, "último en entrar, primero en salir").

NODO:

Es una estructura sencilla que almacena información y además hace referencia a algún otro nodo. De manera ejemplificada es como una carta o un correo electrónico que además del documento en sí, posee la dirección del remitente y la del receptor.

Keywords

GUI:

Commonly known as *Graphical User Interface*, it is a computer program that acts as a user interface, using a set of images and graphical objects to represent the information and actions available in the interface. Its main use is to provide a simple visual environment to allow communication with the operating system of a machine or computer.

Array:

Arrays or as some call them "multidimensional arrays" are a data structure quite similar to vectors or arrays. In fact, an array is nothing more than a series of vectors contained in each other (or others), that is, an array is a vector whose positions are other vectors. Let's talk more about this to be clearer.

Lists:

A list is an ordered collection of values. A list can contain anything. Lists are linear and dynamic data structures. The main advantage of dynamism is the fact that memory positions are acquired as they are needed and released when they are no longer required.

PILA:

This is an ordered list or data structure that allows you to store and retrieve data, being the mode of access to its elements of type LIFO (Last In, First Out).

Node:

It is a simple structure that stores information and also refers to some other node. In an exemplified way it is like a letter or an email that in addition to the document itself, has the sender's address and that of the receiver.

Introducción

Utilizando el paradigma de la *Programación Orientada a Objetos*, las TDA'S (Tipos de Datos Abstractos) y las listas como un medio dinámico de almacenamiento de datos, bajo la comprensión que una imagen esta creada por medio de matrices las cuales guardan la información de cada carácter y/o byte empleado en las imágenes, se puede mencionar que en las TDA se puede almacenar la información para poder realizar una imagen.

Así mismo se menciona que por medio de dichas TDA, se puede realizar operaciones relevantes en las imágenes así como unirlos y encontrar diferencias entre cada punto en común de cada imagen (tomando un sistema de coordenadas x, y) y alteraciones en las imágenes como rotarlas de forma horizontal, vertical, implementar líneas, marcos, etc.

Desarrollo del tema

Tomando como punto de partida el análisis de las imágenes las cuales comúnmente se conoce que están creadas por medio de un conjunto de bytes implementando el alfabeto RGB utilizado en las ciencias de la computación para transformar y dar color a x entidad, y profundizando en dicho análisis se puede mencionar que las imágenes están realizadas por medio de bytes unidos y especificando su almacenaje en una matriz ortogonal la cual puede dejar en blanco o colocar predeterminadamente un color blanco (#FFFFFF) y posteriormente en las posiciones donde si tiene valores (colores) significativos guardar dicha información.

Para profundizar en dicho análisis se puede realizar una implementación del mismo análisis utilizando TDA'S para realizar las matrices ortogonales y con ellas por medio de un GUI (Interfaz Grafica de Usuario) se podrá observar el resultado de dicho

análisis comprobando la aceptación de este; por lo cual se debe de conocer algunos temas al respecto para poder comprender el funcionamiento de este; entre dichos conocimientos se destacaran los siguientes, los cuales se profundizaran posteriormente:

- Programación Orientada a Objetos
 - Clases
 - Métodos
 - Funciones
- Tipos de Datos Abstractos
 - Apuntadores o Punteros
 - Listas enlazadas
 - Pila
- Matriz Dispersa
 - Matriz Ortogonal como un caso especial de la matriz dispersa
- GUI
 - Widget

Así mismo para mostrar un reporte, informe o bitácora de la información que se creo (imágenes) por medio de las matrices ortogonales se deberá de conocer los conceptos de:

- HTML
- CSS
- Maquetación

PROGRAMACION ORIENTADA A OBJETOS:

Es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos se utilizan como metáfora para emular las entidades reales del negocio a modelar. Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Está basada en varias técnicas del sexenio: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.



Ilustración I. Programación Orientada a Objetos

CLASES:

Una clase es una especie de "plantilla" en la que se definen los atributos y métodos predeterminados de un tipo de objeto. Esta plantilla se crea para poder crear objetos fácilmente. Al método de crear nuevos objetos mediante la lectura y recuperación de los atributos y métodos de una clase se le conoce como instanciación.

MÉTODOS:

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

FUNCIONES:

En programación, una función es una sección de un programa que calcula un valor de manera independiente al resto del programa.

Una función tiene tres componentes importantes:

- los **parámetros**, que son los valores que recibe la función como entrada;
- el **código de la función**, que son las operaciones que hace la función; y
- el **resultado** (o **valor de retorno**), que es el valor final que entrega la función.

En esencia, una función es un mini programa. Sus tres componentes son análogos a la entrada, el proceso y la salida de un programa.

TIPOS DE DATOS ABSTRACTOS:

Es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en como estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

El paradigma de orientación a objetos permite el encapsulamiento de los datos y las operaciones mediante la definición de clases e interfaces, lo cual permite ocultar la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

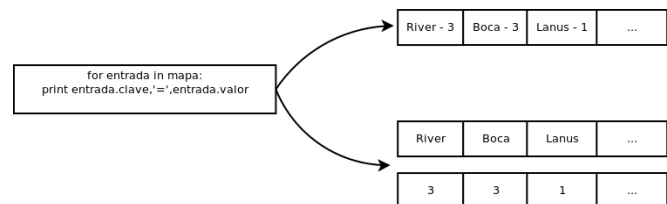


Ilustración II. Tipos de Datos Abstractos

Para conocer mas de los Tipos de Datos abstractos debemos de saber que la mayoría de los Tipos de

Datos Abstractos utilizan un espacio de memoria que lo redirige a otros espacios de memoria llamados Apuntadores, y un espacio de memoria donde se almacena la información; a dicha unión se le conoce como Nodo, el cual se observa en la Ilustración III. (Anonimo, s.f.)

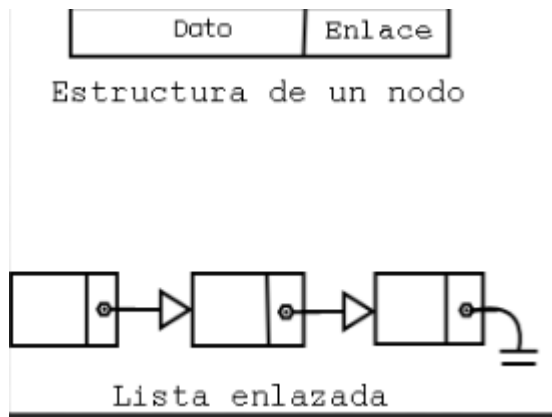


Ilustración III. Estructura de un Nodo

APUNTADORES:

- Los apuntadores son variables que almacenan direcciones de memoria.
- En general una variable contiene un valor específico dependiendo de como fue declarada.
- Un apuntador contiene la dirección de una variable que contiene un valor específico.
- Una variable se refiere directamente a un valor y un apuntador se refiere indirectamente a un valor.
- Apuntadores usados en C debido a que a veces son la única manera de expresar un cálculo.
- Se puede llegar a obtener un código más compacto y eficiente.
- Cuando se emplean sin cuidado pueden crear programas imposibles de entender.
- Cuentan con una declaración propia.
- Los apuntadores disponen de dos operadores: El operador unario o monádico "&" devuelve la dirección de memoria de una variable; El operador de indirección o desreferencia "*" devuelve el contenido de un objeto apuntado por un apuntador".

LISTAS ENLAZADAS:

Una lista se define como una serie de N elementos E_1, E_2, \dots, E_N , ordenados de manera consecutiva, es decir, el elemento E_k (que se denomina elemento k-ésimo) es previo al elemento E_{k+1} . Si la lista contiene 0 elementos se denomina como lista vacía.

Las operaciones que se pueden realizar en la lista son: insertar un elemento en la posición k, borrar el k-ésimo elemento, buscar un elemento dentro de la lista y preguntar si la lista esta vacía.

Una manera simple de implementar una lista es utilizando un arreglo. Sin embargo, las operaciones de inserción y borrado de elementos en arreglos son ineficientes, puesto que para insertar un elemento en la parte media del arreglo es necesario mover todos los elementos que se encuentren delante de él, para hacer espacio, y al borrar un elemento es necesario mover todos los elementos para ocupar el espacio desocupado. Una implementación más eficiente del TDA se logra utilizando listas enlazadas.

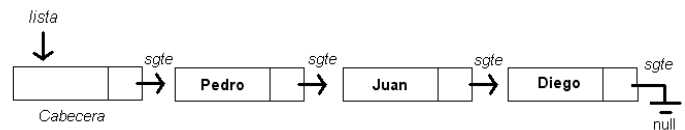


Ilustración IV. Listas Enlazadas

Comúnmente las listas utilizan distintos métodos dependiendo de la información que se manejara y su debida implementación, mas los métodos que frecuentemente se utilizan son los siguientes:

- EstaVacía(): dicho método tiene la funcionalidad de verificar si la cabeza de la lista implementada esta vacia o tiene algún objeto.
- Insertar(): dicho método tiene la funcionalidad de ingresar la información dada, la cual posteriormente se podrán observar y recorrer por medio de los apuntadores de cada Nodo.
- Buscar(): dicho método tiene la funcionalidad de buscar el valor x en la lista implementada.

- Eliminar(): dicho método tiene la finalidad de buscar y eliminar el valor x de la lista implementada.

Así mismo se puede mencionar que dependiendo de la cantidad y comportamiento de los apuntadores de cada nodo se subdividen en los siguientes tipos de Listas:

- ▲ Listas Simplemente enlazadas: la cual cada Nodo tiene un solo apuntador, el cual apunta al nodo siguiente de este.
- ▲ Lista Doblemente Enlazada: en la cual cada Nodo tiene dos apuntadores los cuales uno apunta al Nodo anterior y el otro apunta al Nodo siguiente de este.
- ▲ Lista Circular enlazada: en la cual cada Nodo tiene solo un apuntador el cual apunta o dirige al nodo siguiente, con la distinción es que el ultimo Nodo apuntara a la cabeza de la lista (el primer Nodo).
- ▲ Lista Doblemente Circular: en la cual cada Nodo tiene dos apuntadores los cuales al igual que la Lista Doblemente Enlazada uno apunta al nodo anterior y el otro apunta al nodo siguiente, con la distinción que el primer Nodo apunta al siguiente de este y al ultimo Nodo de la lista; y a su vez el ultimo nodo apunta al nodo anterior de este y al Primer nodo de la Lista.

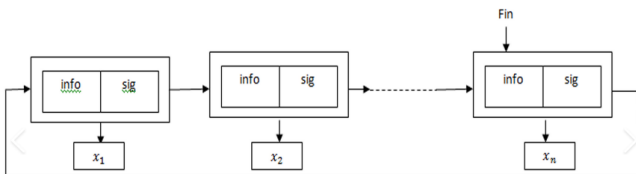


Ilustración V. Lista Circular simple

Así mismo debido a la forma de ingresar y sacar los datos de la lista tiene dos nombres los cuales son:

- Pilas: Una pila (stack o pushdown en inglés) es una lista de elementos de la cual sólo se puede extraer el último elemento insertado. La posición en donde se encuentra dicho elemento se denomina tope de la pila. También se conoce a las pilas como listas

LIFO (LAST IN - FIRST OUT: el último que entra es el primero que sale).

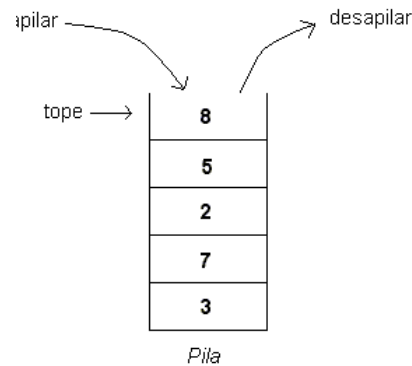


Ilustración VI. TDA Pila

- Colas: Una cola (queue en inglés) es una lista de elementos en donde siempre se insertan nuevos elementos al final de la lista y se extraen elementos desde el inicio de la lista. También se conoce a las colas como listas FIFO (FIRST IN - FIRST OUT: el primero que entra es el primero que sale).

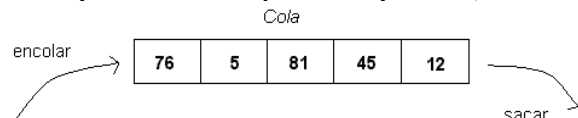


Ilustración VII. TDA Cola

Otras TDA'S que se utilizan debido a la forma necesaria de manejar los datos y de una forma muy específica son las:

- Matrices Dispersas
- Árboles Binarios
- Árboles N-arios

En este caso se profundizará más en las Matrices Dispersas debido al estudio del manejo de datos en la misma.

MATRICES DISPERSAS:

Según la matemática clásica se puede definir una *Matriz Dispersa* se puede decir que es: es una matriz de gran tamaño en la que la mayor parte de sus elementos son cero.

Más en las ciencias de la computación se puede definir la *Matriz Dispersa* como, son matrices en

las cuales existen gran cantidad de valores nulos o ceros. El aprovechamiento de este conocimiento permite reducir el costo computacional de las operaciones que se pueden realizar sobre estas matrices, así como el costo espacial para el almacenamiento de la información. Las matrices dispersas permiten el procesamiento de grandes volúmenes de información de conjuntos de datos no densos. En muchos casos el manejo de estos volúmenes de datos sin tener en cuenta la densidad de estos, se vuelve impracticable, pues operaciones como la multiplicación de matrices, se vuelve un proceso altamente costoso desde el punto de vista computacional, tanto temporal como espacial. Muchas implementaciones sobre este tipo de matrices han sido llevadas a cabo con estructuras como listas y arreglos dinámicos, los cuales mejoran el procesamiento de matrices dispersas, pero no logran aprovechar todas las potencialidades de estas matrices, pues solo tienen en cuenta una sola dimensión dispersa. En este trabajo se propone una implementación de matrices dispersas en el lenguaje de programación Java empleando árboles balanceados para su implementación. Se comparan los resultados con otras implementaciones de matrices dispersas.

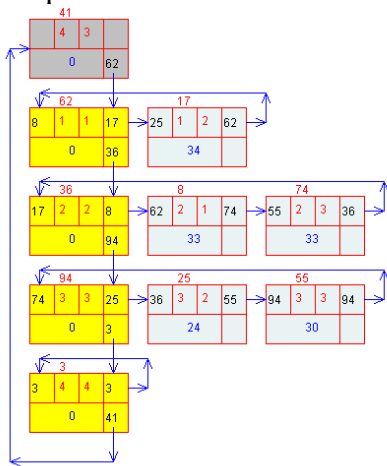


Ilustración VIII. Matriz Dispersa

MATRIZ ORTOGONAL, CASO ESPECIAL DE LA MATRIZ DISPERSA:

Una matriz ortogonal es una estructura de datos que implementa una tabla con memoria dinámica, se

puede buscar o recorrer por uno de los dos aspectos de orden.

La lista de los eventos esta contenida en el nodo de día, este nodo de día esta contenido en una Matriz Ortogonal(en el proyecto es la clase melOrto.java) o Lista Ortogonal, es decir cada nodo tiene 4 punteros a otros nodos. Cada instancia de la Matriz Ortogonal conforma una lista doble enlazada que es con la que se maneja el año en curso.

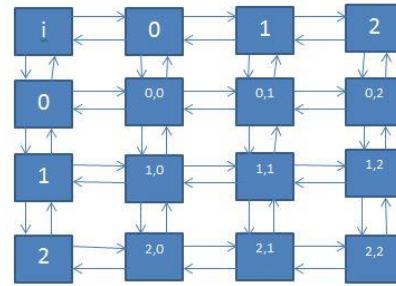


Ilustración IX. Matriz Ortogonal

GUI (Interfaz gráfica de usuario):

La interfaz gráfica de usuario, conocida también como GUI (del inglés graphical user interface), es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

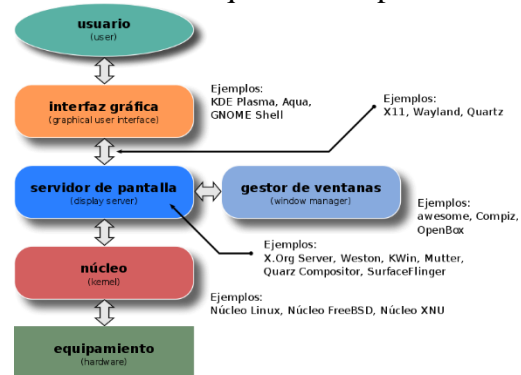


Ilustración X. GUI (Interfaz Gráfica de Usuario)

WIDGET:

Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget

Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. En el Lenguaje de Programación Python a cada una de las partes de un GUI se le considera Widget entre ellas:

- Form
- Frame
- Label
- Button
- Etc.

HTML:

“HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto”.¹

En palabras más sencillas para el espectador HTML es un lenguaje marcado el cual tiene la función de poder realizar páginas web, dicho lenguaje está creado y confeccionado por medio de etiquetas y un tipo de maquetado específico para lo cual se debe de saber principalmente que es una etiqueta ya que es la base de dicho lenguaje.

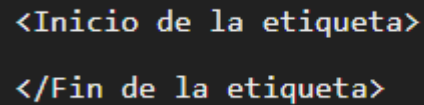
ETIQUETAS:

Es la estructura principal del lenguaje HTML, marcas de hipertexto y son las encargadas de organizar el contenido de las páginas web, también conocidas como “tags”. Entre las etiquetas que se pueden destacar y mencionar que son las más importantes se puede decir:

- Html
- Head
- Body
- Title
- Etc.

Ahora que entendemos el concepto de etiquetas, necesitamos poder identificarlas. Éstas se caracterizan porque van dentro de los caracteres menor que “<” y mayor que “>”, como en este ejemplo: “<Aquí dentro va el nombre de la etiqueta>”. Así mismo las etiquetas tienen una

estructura definida la cual tiene que ser respetada para ser considerada con etiqueta la cual se observa en la *Ilustración XI*. (Herrera, 2015)



```
<Inicio de la etiqueta>
</Fin de la etiqueta>
```

Ilustración XI. Estructura de las etiquetas

CSS:

CSS (en inglés Cascading Style Sheets) es lo que se denomina lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS separa el contenido de la representación visual del sitio.

El lenguaje CSS se puede implementar de 3 maneras diferentes las cuales son:

- **Manera Interna:** Los estilos CSS hechos de esta manera se cargan cada vez que se actualiza un sitio web, lo que puede aumentar el tiempo de carga. Además, no podrás usar el mismo estilo CSS en varias páginas, ya que está contenido en una sola página. Sin embargo, esto también tiene sus beneficios.
- **Manera Externa:** Todo se hace externamente en un archivo .css. Esto significa que puedes hacer todo el estilizado en un archivo separado y aplicar el CSS a cualquier página que quieras. El estilo Externo también puede mejorar los tiempos de carga.
- **Manera inline:** Inline trabaja con elementos específicos que tienen la etiqueta <style>. Cada componente tiene que ser estilizado, por lo que podría no ser la mejor forma, ni la más rápida para manejar CSS. Pero puede ser útil, por ejemplo, si quieres cambiar un solo elemento, tener una vista previa rápida de los cambios o tal vez no tengas acceso a los archivos CSS. (Gustavo, 2019)

¹ Código Facilito, ¿Qué es HTML?

Selectores en CSS:

Los selectores CSS se utilizan para "buscar" (o seleccionar) los elementos HTML que desea aplicar estilo.

En CSS se conocen varios tipos de selectores mas principalmente se conocen los siguientes selectores:

- El selector de elementos CSS: El selector de elementos selecciona elementos HTML en función del nombre del elemento.

```
p {  
  text-align: center;  
  color: red;  
}
```

Ilustración XII. Selector de elementos

- El selector CSS id: El selector id utiliza el atributo id de un elemento HTML para seleccionar un elemento específico. ¡El identificador de un elemento es único dentro de una página, por lo que el selector de id se utiliza para seleccionar un elemento único! Para seleccionar un elemento con un identificador específico, escriba un carácter hash (#), seguido del identificador del elemento.

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

Ilustración XIII. Selector Id

- El selector de clases CSS: El selector de clases selecciona elementos HTML con un atributo de clase específico. Para seleccionar elementos con una clase específica, escriba un carácter de punto (.), seguido del nombre de la clase.

```
.center {  
  text-align: center;  
  color: red;  
}
```

Ilustración XIV. Selector class

- El selector universal CSS: El selector universal (*) selecciona todos los elementos HTML de la página.

```
* {  
  text-align: center;  
  color: blue;  
}
```

Ilustración XV. Selector Universal

(w3schools.com, s.f.)

Referencias

- Anonimo. (s.f.). *users.dcc.uchile.cl*. Obtenido de <https://users.dcc.uchile.cl/~bebustos/apuntes/c30a/TDA/#:~:text=de%20datos%20abstractos-,Tipos%20de%20datos%20abstractos,como%20est%C3%A9n%20implementadas%20dichas%20operaciones>.
- Gustavo. (13 de mayo de 2019). *Hostinger Tutoriales*. Obtenido de <https://www.hostinger.mx/tutoriales/que-es-css>
- Herrera, J. F. (25 de Agosto de 2015). *Codigo Facilito*. Obtenido de <https://codigofacilito.com/articulos/que-es-html>
- w3schools.com. (s.f.). *w3schools.com*. Obtenido de <https://www.w3schools.com/css/default.asp>