

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE CIENCIAS DE LA COMPUTACIÓN

PRIMER PROYECTO DE ESTRUCTURA DE DATOS

IMPLEMENTACIÓN DE JUEGO “EL SOLITARIO”



GRUPO N° 1

INTEGRANTES:

- FERRO PALOMINO, GIANFRANCO AUGUSTO
- IPANAQUE CASQUINA, INGRID FRANSHESKA
- HIDALGO CHAVEZ, DANIEL ALFREDO
- HUACAC CHIRI, CLAUDIA ESTEFFANI
- AGUIRRE JANAMPA, CRISTIAN FERNANDO

INTRODUCCIÓN

La programación estructurada desempeña un papel de elevada importancia en la informática, también en el aprendizaje de todo programador y como toda disciplina presenta un sinfín de problemas que por si mismos suponen un desafío en el avance de la misma. Por lo cual requiere de cualidades indispensables como la lógica, abstracción, la formalización y el orden, las cuales ayudan a discernir los conceptos e ideas.

La realización de este proyecto se nos es necesario para aplicar y tener una mayor aprehensión de los conocimientos aprendidos en el transcurso de las clases.

El siguiente proyecto que mostraremos a continuación está basado en el lenguaje de programación C, ya que éste lenguaje es multiplataforma y funciona en diferentes sistemas operativos, donde pondremos en práctica conceptos como estructuras, arreglos, pilas y muchos conceptos más aprendidos a lo largo de nuestras clases necesarios para la implementación de nuestro proyecto: juego “El solitario “.

Al utilizar arreglos para implementar pilas se tiene la limitación de que se debe reservar el espacio en memoria con anticipación. Una vez dado un máximo de capacidad a la pila no es posible insertar un número de elementos mayor que el máximo establecido. Si esto ocurre, en otras palabras si la pila esta llena y se intenta insertar un nuevo elemento, se producirá un error conocido como desbordamiento –overflow

OBJETIVOS

- Crear un programa del juego “El solitario” en el entorno de desarrollo de ubuntu.
- Desarrollo y aplicación de conocimientos de estructuras, arreglos y pilas en lenguaje C.
- Posible desarrollo de una interfaz gráfica para el proyecto .
- Explicar de forma detallada el programa y la implementación a realizarse.
- Obtener nuevos conceptos de la estructura de datos a través de la investigación personal y grupal.
- Aprender a trabajar en equipo para lograr un mejor aprendizaje, organización y comunicación.

DESARROLLO

El desarrollo del siguiente proyecto se definirá de la siguiente manera:

La aplicación que se desarrollará en el lenguaje de programación C, es la simulación de un juego llamado “El solitario” en C, el cual tendrá ciertas características que iremos detallando:

El juego consta de una baraja de 52 cartas de las cuales distribuiremos 28 de las cartas en 7 pilas de juego.



Las cartas restantes son colocadas cerradas en otra pila llamada “pila de reserva”.

Existe una pila de “descarte” y 4 pilas de “salida”.

En total serán 13 pilas.

MARCO TEÓRICO

- Nodos

Una lista se compone de nodos, que son estructuras de datos que nos permiten registrar datos de [interés](#). Para que estos nodos se conviertan en una lista, debe existir un enlace entre ellos, que en términos más propios se los conoce como apuntadores o punteros. Los punteros o apuntadores, son la parte de los nodos que nos permiten recorrer la lista y acceder también a las direcciones de memoria donde se almacenan los elementos de la lista. Como se muestra en

la Figura 1, La lista tiene un *inicio* y tiene un *fin*. El inicio se lo conoce como *cabecera*, mientras que el final de lista se lo conoce como *cola*. Para poder darle fin a la lista, lo único que se hace es que el último apuntador o último enlace apunte a *NULL*, es decir, que apunte a nada. Para acceder a un elemento de la lista simplemente enlazada, esta debe ser recorrida comenzando por el inicio (o cabecera como se mencionó); el puntero siguiente o el enlace, permite recorrer o desplazarnos hacia el próximo elemento, hasta llegar al final.

En el caso de las listas simplemente enlazadas, el desplazamiento se hace en una sola dirección, es decir, que el apuntador o enlace de un nodo apunta solo al nodo inmediato siguiente, desde el primer nodo (cabecera), hasta el último nodo (cola).

Si es que se deseara desplazarse tanto hacia adelante, como hacia atrás, en las listas simplemente enlazadas no se puede hacer eso, y tampoco hay **métodos** o funciones que nos permitan hacer eso. Para conseguirlo, la única alternativa es usar listas doblemente enlazadas.

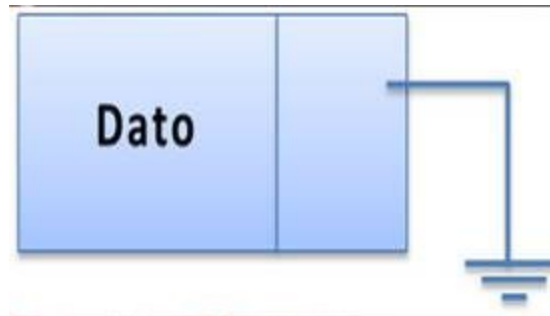


Figura 2. Un nodo se compone de dos partes, la primera es la parte de *Dato*, que es donde se registran datos de interés; y la segunda parte que es la *Dirección*, que es la parte que nos ayuda a crear enlaces entre nodos y así generar listas.

CONCLUSIONES

REFERENCIAS

- The art of computer programming, Knuth Donald, editorial: Addison-Wesley vol. 1, pág. 245,246.

