

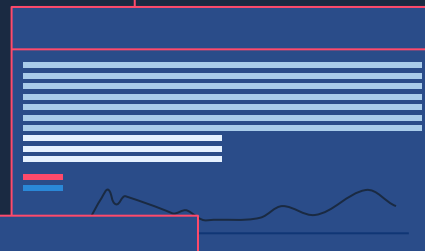


Tecnicatura en Programación Universitaria

Comisión

**B**

# Clase 06

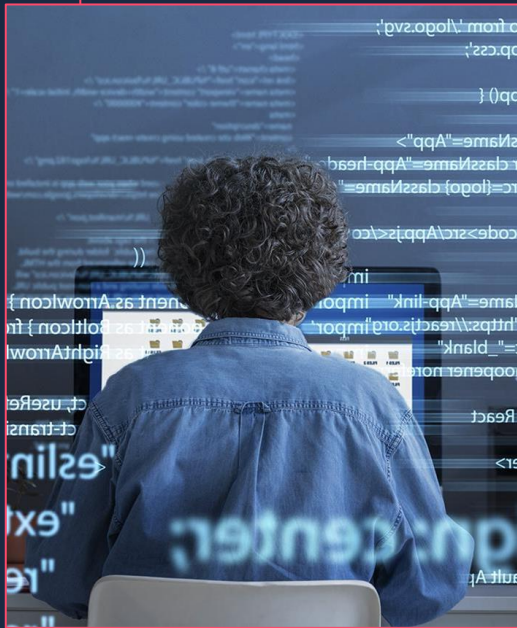


**Clase Practica**

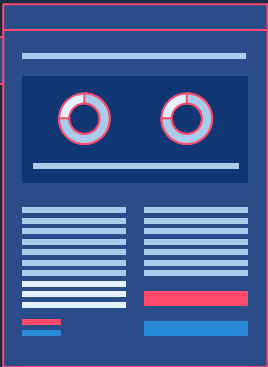
Martes 14 de Mayo 2024

Ing. Oviedo Codigoni Carlos Nicolas





< Numeros Random >



# rand()

La función rand() es una función en C que genera números pseudoaleatorios. Estos números no son realmente aleatorios en el sentido de que son deterministas y dependen del estado inicial de la generación de números aleatorios. Sin embargo, para muchos propósitos prácticos, la secuencia de números que produce rand() parece aleatoria.

La función time(NULL) en C devuelve el tiempo actual del sistema como un valor de tipo time\_t. Este valor representa el número de segundos transcurridos desde una fecha de referencia específica (por lo general, el 1 de enero de 1970, conocida como "Epoch").

```
1  #include <stdlib.h>
2  #include <time.h>
3
4  int main() {
5      // Establece la semilla utilizando el tiempo actual
6      srand(time(NULL));
7      int numAleatorio = rand();
8      // Genera un número pseudoaleatorio entre 0 y RAND_MAX
9      return 0;
10 }
```

La secuencia de números generada por rand() depende de un valor inicial llamado semilla (seed). Por defecto, la semilla es el tiempo actual del sistema, lo que hace que la secuencia parezca aleatoria. Sin embargo, puedes establecer manualmente la semilla usando la función srand().

Cuando utilizas time(NULL), estás obteniendo el tiempo actual en segundos desde el Epoch, y este valor se utiliza a menudo para establecer la semilla (seed) en la función srand() para inicializar la generación de números pseudoaleatorios. Al usar time(NULL) como argumento para srand(), se garantiza que cada vez que ejecutes el programa, la semilla será diferente, lo que ayuda a producir secuencias pseudoaleatorias más variadas.

## # include<exercise.h>

**Adivinar el número secreto:** El programa elige un número aleatorio y le pide al usuario que lo adivine. Después de cada intento, el programa proporciona pistas al usuario si el número ingresado es mayor o menor que el número secreto. Generar  $n$  intentos y guardar en un vector si acierta en cada intento.

**Ahorcado:** El programa selecciona aleatoriamente una palabra de una lista predefinida y le pide al usuario que adivine la palabra letra por letra. El usuario tiene un número limitado de intentos para adivinar la palabra completa.

**Batalla Naval:** El programa simula el juego de Batalla Naval, donde el usuario coloca barcos en un tablero y luego intenta adivinar las ubicaciones de los barcos del oponente. El programa lleva la cuenta de los barcos hundidos y el usuario intenta hundir todos los barcos del oponente en la menor cantidad de intentos posible.

**Piedra, papel o tijera:** El programa le pide al usuario que elija entre piedra, papel o tijera, y luego genera una elección aleatoria para la computadora. El programa determina quién gana según las reglas del juego.

**Crucigrama:** El programa genera un crucigrama aleatorio y le pide al usuario que complete las palabras faltantes. Puedes usar matrices para representar el tablero del crucigrama y vectores para almacenar las palabras y sus ubicaciones.

**Buscador de palabras en sopa de letras:** El programa genera una sopa de letras y le pide al usuario que encuentre palabras específicas en ella. Puedes usar matrices para representar la sopa de letras, vectores para almacenar las palabras que el usuario debe encontrar y funciones para buscar las palabras en la matriz.