**Universidad Tecnológica Centroamericana**

**UNITEC**

****

Semestre 2, Periodo IV

Laboratorio de Programación II

Alberto Josué Melgar Rivera

Laurenn Alecxandra Cruz

**Laboratorio N° 8**

# Objetivos

Entender la lectura y escritura de archivos binarios.

Hacer uso de hilos.

# Desarrollo de la práctica

El profesor Elm necesita que usted desarrolle una aplicación GUI en Java que simule una aventura en el safari pokémon. La aplicación requiere las siguientes funcionalidades:

* Leer Objetos y Pokémons de los archivos (Objetos.txt y Pokemons.txt).
* Mostrar los pokémon atrapados (Reporte de los datos del Pokedex, mostrar en JTable).
* Listar (en JTable) objetos de la mochila.
* Guardar.
* Pescar.
* Buscar (aleatoriamente):
  + Pokémon en grama
  + Objetos

Antes de iniciar el *simulador de safari pokémon,* se deben leer los pokémon y objetos de los archivos de texto y guardarlos en dos ArrayList. Una vez leídos los pokémon y objetos se habilita el botón ***iniciar simulador***.

Al iniciar el *simulador de safari pokémon* sucede lo siguiente:

1. Se comprueba si el usuario ya tenía objetos en la mochila (leer los objetos del archivo binario Mochila.bin) y mostrar un mensaje: “La mochila se ha llenado”.
2. Se comprueba si el usuario ya tenía pokémons registrados en el pokedex (leer los pokémons del archivo binario Pokedex.bin) y mostrar un mensaje: “El pokedex se ha actualizado”.
3. 30 safariBalls se agregan a la mochila y mostrar un mensaje:”30 Safari balls se han agregado a la mochila”. La safariBall es un Objeto.
4. Habilitar las opciones: buscar pokémon, buscar Objetos, reporte de Objetos, ver pokedex, guardar, pescar.

* Para la opción ***buscar Objetos*** se agrega aleatoriamente un Objeto a la mochila. Si la mochila contiene ese objeto la cantidad de ese Objeto se incrementa.
* Para la opción ***buscar un pokémon en grama*** se abre una ventana con un JTextField con el nombre del pokémon, otro JTextField que muestre el HP actual del pokémon de la forma (HP actual/HP total), otro JTextField que muestre el nivel, un JTextArea para los mensajes de la batalla el cual se inicializa con el mensaje “Nombre\_de\_Pokemon salvaje, ha aparecido” y 4 botones:

1. **Lanzar roca**: En esta opción se le resta 20% del total de HP a los HP del pokémon. Si HP llega a 0 la batalla finaliza Y se Agrega al JTextArea el mensaje:”Nombre\_de\_Pokemon, se debilitó”
2. **Lanzar comida**: En esta opción se le suma 5% del total de HP a los HP del pokémon, y se le asigna a BE 2. BE se debe inicializar con valor 1.
3. **Lanzar safariBall**: En esta opción se calcula ***a***:

.

Si a>=255, el pokémon es capturado. Si no es así se calcula ***b***.

Entonces se le asignan valores aleatorios entre el 0 y el 65535 a cuatro números: n1, n2, n3 y n4.

Si ***b*** es mayor o igual a 1 de los números aleatorios o menor a todos, se debe agregar al JTextArea el mensaje: *“Oh no! El pokémon se liberó!\n”.*

Si ***b*** es mayor o igual a 2 de los números aleatorios se debe agregar al JTextArea el mensaje: *“Casi lo tenía\n”.*

Si ***b*** es mayor o igual a 3 de los números aleatorios se debe agregar al JTextArea el mensaje: *“Parecía estar atrapado!\n”.*

Si ***b*** es mayor o igual a los 4 números aleatorios el pokémon será atrapado y se debe agregar al JTextArea el mensaje: *“Nombre\_de\_Pokémon fue atrapado\n”.*

**Cada vez que se lanza una safariBall, la cantidad de safariBalls en la mochila debe decrementar.**

1. **Huir**: Salir de la batalla.

Después de seleccionar un botón, el pokémon reaccionara de diferentes maneras agregando al JTextArea aleatoriamente cualquiera de los mensajes:

”Nombre\_de\_Pokémon parece tener confianza en usted\n”,

”Nombre\_de\_Pokémon se ha asustado\n”,

”Nombre\_de\_Pokémon usó Nombre\_de\_Ataque, pero usted lo esquivó\n”.

Cada batalla debe guardar lo que contiene el JTextArea además de la fecha con hora actual en un Archivo llamado batalla\_N.txt. N será el número de batalla, el cual incrementa cada vez que se registra una batalla.

* Al momento de ***guardar***, se guardan los datos de las listas mochila y pokedex en los archivos binarios correspondientes.
* Para la opción ***pescar***, se debe abrir una nueva ventana con un JTextArea, inicia un hilo el cual escribirá ***n*** puntos en el textArea, después de terminar de escribir los puntos escribe “Oh! ha mordido”, el hilo hace sleep por 1000 milisegundos y en ese momento el usuario debe presionar la barra espaciadora para jalar de la caña, si no presiona la barra espaciadora en ese momento o si presiona la barra espaciadora mientras está escribiendo los puntos, escribe el mensaje “Ha huido”, este procedimiento se realiza ***x*** veces. Los números ***“x”*** y ***“n”*** son aleatorios entre 1 y 5. Si logra jalar de la caña ***x*** veces, se inicia una batalla con un pokémon de tipo agua.

Deberá hacer uso de las clases:

**Pokémon**:

* Nombre: String
* Nivel: int
* Ataque: int
* Defensa: int
* Velocidad: int
* Ataque Especial: int
* Defensa Especial: int
* HP: int
* Tipo1: String
* Tipo2: String
* Ataques: ArrayList<Ataque> de 4
* rareza(catching ratio): int (0-255)
* Sprite: ImageIcon

El nombre se lee del archivo Pokémons.txt

El nivel es un número aleatorio entre 20 y 100.

Ataque, defensa, velocidad, ataque especial, defensa especial, HP se calculan con la misma fórmula:

Donde “x” es un numero aleatorio entre 2 y 4.

Tipo1 y tipo2 se leen desde el archivo Pokémons.txt

Los Ataques se agregan al ArrayList de manera aleatoria entre los ataques leídos desde el archivo Ataques.txt. **Ojo**, solo son 4 ataques, el pokémon no puede tener ataques repetidos y se debe validar que los ataques sean del mismo tipo del pokémon, ejemplo si el pokémon es de tipo1 es veneno y tipo2 es tierra entonces los ataques tienen que ser de tipo veneno o tierra.

La rareza se lee desde el archivo Pokémons.txt. El sprite se lee de un archivo de tipo png.

**Ataque**:

* Nombre: String
* Tipo: String
* PP: int
* Poder: int
* Precision: int

Todos estos atributos se leen desde el archivo Ataques.txt

**Objeto**:

* Nombre: String
* Cantidad: int

Todos estos atributos se leen desde el archivo Objetos.txt

# Evaluación

Lectura de Archivos Binarios 1.5%

Escritura de Archivos Binarios 1.5%

Clases 1%

Reportes 2%

Lectura y Escritura en Archivos de texto 1%. 0.25% c/u.

Simulación de batalla 1.5%

Pescar 1.5%

# Entrega

Deberá mandar un comprimido en extensión .ZIP, .RAR o .TAR.BZ2 con la siguiente información:

[NOMBRE\_COMPLETO]\_[NUMERO\_DE\_CUENTA]\_[NUMERO\_DE\_LABORATORIO].[EXTENSION]

Tiene hasta las 5:00 pm de hoy para desarrollar el laboratorio, y subirlo.