

# Universidad de San Carlos de Guatemala

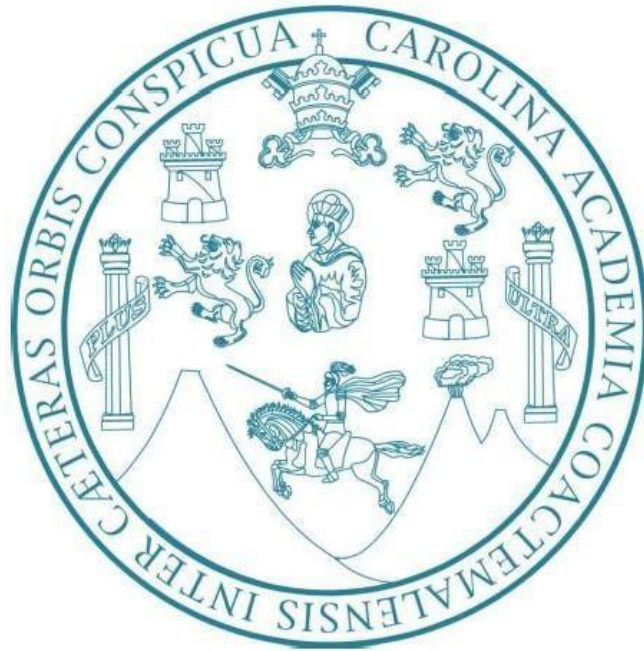
Facultad de Ingeniería

Introducción a la Programación y Computación 1

Sección: N

Ing. Neftali De Jesus Calderon Mendez

Tutor académico: Jurgen Ramirez



## PRÁCTICA 1

JUEGO EN CONSOLA

*Desarrollo de juego con temática "Batalla Pokémon".*

# TABLA DE CONTENIDO

## Contenido

Objetivos_____	1
Descripción General_____	2
Requerimientos para el desarrollo del proyecto:_____	4

# PRACTICA 1 - IPC1

## Objetivos

### GENERALES

- Familiarizar al estudiante con el lenguaje de programación JAVA.
- Que el estudiante aplique los conocimientos adquiridos en el curso de Introducción a la Programación y computación 1.
- Elaborar la lógica para presentar una solución a la propuesta planteada.

### ESPECÍFICOS

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software.
- Construcción de aplicaciones simples en consola.
- Implementación de sentencias de control, ciclos y vectores.
- Aplicación de programación estructurada
- Aplicación de conceptos de diagramas de flujo, diagramas de clases, ciclo de vida de software.

# PRACTICA 1 - IPC1

## Descripción General

### DEFINICIÓN:

La práctica consiste en el desarrollo de una aplicación de consola que simulará un juego de batalla entre 2 jugadores, donde cada personaje estará representado por figuras del famoso anime Pokémon. La interacción del juego con el usuario será de manera sencilla mediante un menú en consola.

### MENÚ

Se debe de poder ingresar como usuario **normal** y **administrador**. El menú de administrador y usuario normal debe de tener un menú donde se puedan tener las siguientes opciones.

#### Menú principal

- Ingresar al sistema
  - **Administrador (La contraseña debe de ser su carnet)**
  - **Iniciar juego con dos jugadores** (A los jugadores se les debe de pedir se les debe de pedir su nombre)

#### Salir

#### Administrador

- **Lista de Pokémon**
- **Agregar Pokémon**
- **Editar Pokémon**
- **Reportes**
- **Regresar al menú principal**

#### Iniciar Juego con dos jugadores

Al ingresar a este menú se les debe de pedir los **dos nombres** de los jugadores y luego solicitarle a cada jugador los **dos Pokémon** con los que jugaran. Se les debe de **mostrar el listado de Pokémon** con todas sus características para que puedan elegir.

- **Opción de batallar** (Si no han seleccionado sus Pokémon no podrán realizar la batalla).
- **Regresar al menú principal.**

### POKÉMONS:

- El programa debe de poder almacenar como **mínimo 6 Pokémon y un máximo de 20.**
- Cada pokémon tendrá un **nombre** único, un contador de **vida**, un estado actual (vivo o muerto), una ponderación de **ataque** y una **imagen de caracteres** representativa. Cada uno de los elementos anteriores deberá mostrarse en consola en todo momento únicamente cuando se esté simulando la batalla.
- Se deberán utilizar imágenes representativas de calidad. (ejemplos generados en <https://manytools.org/hacker-tools/convert-images-to-ascii-art/>)



- Los **puntos de vida** de un pokémon **disminuirán al recibir un ataque** durante la batalla, pero al **iniciar** una nueva batalla **se restaurarán las vidas de cada pokémon** que combatió.
- El programa continuara con la ejecución hasta que se seleccione la opción salir.
- La interacción del usuario con la aplicación será mediante la lectura del teclado.

Los datos de los pokémon que se describen a continuación **deben ser generados de manera aleatoria** por la aplicación, el resto pueden ser definidos por el estudiante.

- **Puntos de vida: 50 – 100**
- **Puntos de ataque: 5 – 20**

# PRACTICA 1 - IPC1

## Administrador:

Se debe contar con las siguientes funcionalidades.

- **Funciones principales**
  - **Listar a todos los Pokémon:** al seleccionar esta opción se desplegará en pantalla la información de cada pokémon (imagen, nombre, puntos de vida y puntos de ataque).
  - **Agregar Pokémon:** El administrador podrá agregar nuevos Pokémon al sistema. Se les debe de pedir los datos correspondientes para poder almacenar al Pokémon.
  - **Editar Pokémon:** el administrador podrá editar el nombre, puntos de vida y ataque de cualquier pokémon.
- **Reportes:**
  - **Registro de partidas:** se mostrará un historial de todas las partidas jugadas con el nombre de jugador 1, nombre de jugador 2, lista de pokémon utilizados por cada jugador, número de ataques realizados, número de golpes recibidos de cada pokémon y el ganador de la batalla.
  - **Lista de los pokémon más utilizados:** desplegará en pantalla el nombre y la imagen de los pokémon más utilizados, basado en el número de veces que dichos pokémon han sido seleccionados por los jugadores, este número también debe mostrarse.
  - **Lista de los Pokémon menos utilizados:** desplegará en pantalla el nombre y la imagen de los pokémon menos utilizados, en base a el número de veces que dichos pokémon fueron seleccionados por los jugadores, este número también debe mostrarse.
  - **Jugadores con quienes han perdido:** Se debe de mostrar la lista de jugadores y se debe de poder elegir uno de los jugadores y mostrar cuantas veces ha perdido y con quienes ha perdido.
  - **Jugadores con quienes han ganado:** Se debe de mostrar la lista de jugadores y se debe de poder elegir uno de los jugadores y mostrar cuantas veces ha ganado y con quienes ha ganado.
- **Regresar al menú principal:** Regresara al menú principal.

## BATALLA (Usuarios normales o jugadores):

La batalla se realizará con 2 jugadores, cada uno de ellos podrá elegir 2 pokémon a utilizar durante la partida, se manejarán turnos donde el jugador podrá elegir cual es el pokémon que utilizará para atacar y cuál es el pokémon del rival que desea atacar, luego de haber realizado el ataque se deben disminuir los puntos de vida del pokémon que fue atacado y el turno finalizara, se continuara así sucesivamente hasta que uno de los 2 jugadores tenga a sus 2 pokémon en estado "muerto" (vida 0). La aplicación debe ser capaz de realizar el número de batallas que los jugadores deseen, con distintos nombres de jugador y distintos pokémon. El turno inicial debe ser asignado aleatoriamente entre los jugadores. Durante el transcurso de la batalla se deben visualizar los siguientes datos, nombre e imagen de los pokémon de cada jugador, puntos de vida, puntos de ataque, estado actual (vivo o muerto), nombres de los jugadores, turno del jugador (jugador activo).

Al momento de que finalice la batalla, se debe de preguntar si se desea volver a tener una batalla o salir al menú principal.

Si desean volver a tener una batalla el sistema deberá de solicitar de nuevo a los jugadores que elijan a los dos Pokémon y empezará la batalla. Cada vez que se inicie una batalla la vida de los Pokémon deberán de restablecerse.

## EJEMPLO:

Se plantea una propuesta para mostrar los datos solicitados durante la batalla.

# PRACTICA 1 - IPC1

JUGADOR1: mario



**Nombre:** bulbasaur  
**Vida:** 75  
**Ataque:** 5  
**Estado:** vivo

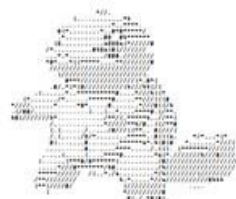


**Nombre:** charmander  
**Vida:** 65  
**Ataque:** 10  
**Estado:** vivo

**TURNO:** jugador1



**Nombre:** bulbasaur  
**Vida:** 75  
**Ataque:** 5  
**Estado:** vivo



**Nombre:** squirtle  
**Vida:** 100  
**Ataque:** 15  
**Estado:** vivo

JUGADOR2: luigui

## Requerimientos para el desarrollo de la practica:

### DOCUMENTACION:

- Diagrama de flujo general del programa.
- Manual Técnico (descripción de los métodos creado) en PDF.
  - Introducción
  - Plataforma de ejecución
  - Diccionario de métodos usados

### RESTRICCIONES:

- La interacción entre el sistema y el usuario será por medio del teclado.
- Para todas las entradas de teclado en consola se debe de validar lo siguiente:
  - Que no acepte dejarlo en blanco.
  - Solo debe de aceptar el tipo de dato que se solicita, por ejemplo, si solicita algún número no se deberá de poder ingresar letras.
- Para poder acceder a los elementos de las **matrices o arreglos** NO se deben de hacer con posiciones específicas matriz [0][1] o matriz [2][3]. Se debe de recorrer la matriz o arreglo con cualquiera de los ciclos de programación for, while o do-while
- La práctica se debe de realizar en forma individual.
- Queda prohibido utilizar las clases Vector, LinkedList, ArrayList o cualquier otra librería dinámica de

# PRACTICA 1 - IPC1

vectores que no haya sido implementada por el estudiante.

- La calificación se realizará sobre el ejecutable (JAR) que se envíe, no se permitirá en la calificación ejecutar desde el IDE
- Para el desarrollo del proyecto se debe utilizar el entorno de desarrollo JAVA en cualquier versión y se puede utilizar cualquier IDE de desarrollo como lo por ejemplo NetBeans, Eclipse u otro.
- **Entregas fuera de tiempo se penalizarán con 40% sobre la calificación.**
- Copias parciales o totales serán sancionadas con cero puntos y reporte a la escuela de Ciencias y Sistemas.
- La Calificación se realizará sobre lo que se envía.

## FORMATO DE ENTREGA:

Adjuntar la documentación, el JAR de la aplicación y la carpeta del proyecto completo en un archivo .rar con el siguiente formato: [IPC1]Practica1\_#carnet.rar por ejemplo: [IPC1]Practica1\_201905822.rar

Subir el archivo .rar en UEDI.

## FECHA DE ENTREGA:

**12 de junio antes de las 23:59 horas.**