Módulo 1: Introducción y conceptos iniciales

1.1 Introducción

Las aplicaciones software hoy en día tienen altas exigencias; millones de usuarios se conectan diariamente, sea en teléfonos móviles, computadoras personales, tabletas, televisores, relojes, etc. A medida crece su popularidad, las plataformas requieren de cierta evolución a través del tiempo, cambios de requerimientos y rotación de los equipos de trabajo. Las exigencias en los equipos de desarrollo crecen día a día.

Todos los anteriores factores, obliga a la ingeniería a crear procesos ordenados, eficientes y creativos que motiven la creación de soluciones robustas y profesionales, que sean de fácil adaptación y que cuente con la documentación requerida.

El paradigma de **programación orientada a objetos** soluciona y cubre algunos de esos requerimientos, conocer y hacer uso de todas sus técnicas para desarrollar software de calidad, que pueda evolucionar fácilmente, que tenga un mantenimiento rápido y por ende, trabajos de entrada y salida innovadores.

Aprenderemos en el curso, todos los conocimientos necesarios para darle forma al paradigma, aplicaremos de manera práctica cada uno de los conceptos en tareas simples y complejas. Conoceremos también cada una de las características que rigen la programación orientada a objetos y su forma adecuada de aplicación que nos permitirá crear aplicaciones de software ordenadas y con código de fácil reutilización.

Los **objetos**, es el termino fundamental de la clase, compuesto por propiedades (Una característica especial de los objetos), relaciones y métodos, es decir, las acciones o funciones que pueden tener los objetos, que expresan su comportamiento, con una estructura definida y datos a los cuales vamos a aprender a acceder.

El concepto de ***clases*** aparece también, donde definimos principalmente las propiedades y como se van a comportar los objetos, aprenderemos a definirlas y usar todas las diferentes características coomo la abstracción, encapsulamiento, polimorfismo, herencia, etc.

En primera instancia, nos encargamos de aprender las principales características de los lenguajes de programación que estaremos usando durante la clase, su estructura y alcance.

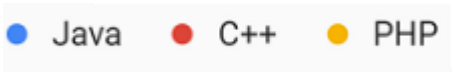
1.2 Definición principal

La programación orientada a objetos "es una de las técnicas más modernas que trata de disminuir el coste del software, aumentando la eficiencia en la programación y reduciendo el tiempo necesario para el desarrollo de una aplicación. Con la programación Orientada a Objetos, los programas tienen menos líneas de código, menos sentencias de bifurcación y módulos que son comprensibles ya que contienen definiciones más claras de las relaciones que existen entre cada uno de los objetos, la reutilización de código se vuelve importante, por lo que se requiere dedicar tiempo necesario para el análisis y diseño del mismo, para que los resultados sean eficientes".

1.3 Lenguajes de programación

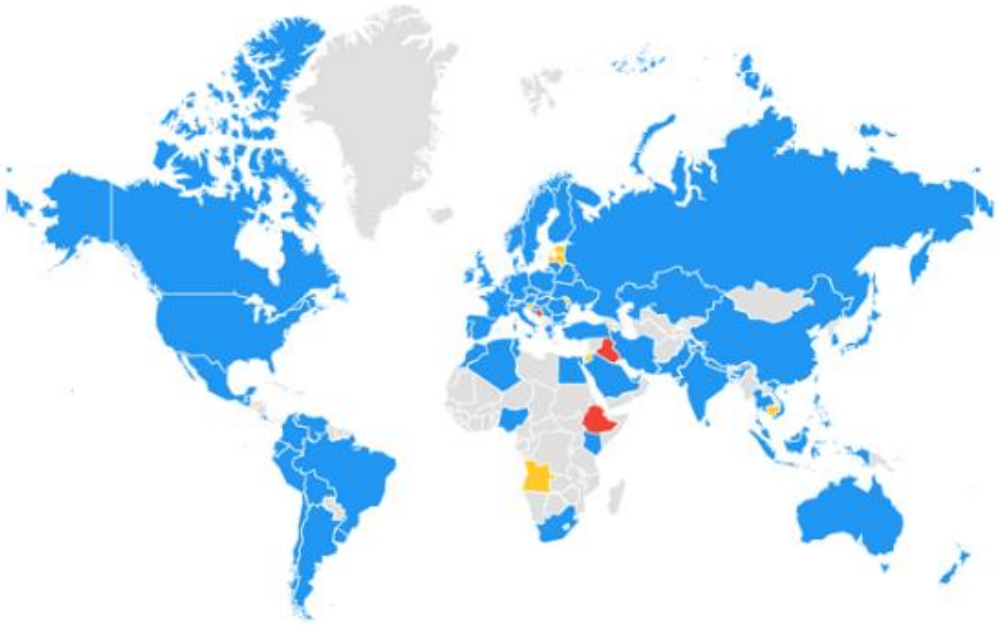
Podemos escribir programas orientados a objetos en muchísimos lenguajes, incluidos los siguientes: Java, C sharp, Ruby, Objetive C, Python, etc. Para fines del curso Programación Orientada a Objetos, trabajaremos principalmente en Java y PHP, dos de los lenguajes más utilizados para construir grandes plataformas a nivel mundial. Lo fundamental es que queden formados los conceptos de POO y que se puedan aplicar en estos lenguajes, que puedan ayudar al estudiante a ser competitivo en sus empresas, y principalmente que puedan emprender con plataformas innovadoras.

En las tendencias de Google, encontramos la popularidad de algunos lenguajes de programación, encontramos que desde hace unos años la constante se mantiene, comparando Java, C++ y PHP.



La tendencia a lo largo de estos últimos años se mantiene, los lenguajes de programación mantienen una constante de uso, Java, lidera en ésta comparación específica.

En términos de región, entre estos tres lenguajes, Java lidera totalmente, es el más buscado de éstos tres en el mundo, lo que demuestra que su uso se extiende por muchas regiones. Vemos en color azul las búsquedas relacionadas con Java, en color rojo para C++ y amarillo para PHP. Si vemos, Honduras no es tomado en cuenta en esta tendencia de Google, necesitamos avanzar los temas de programación de aplicaciones en comparación con algunos países del área centroamericana como Costa Rica, que nos lleva muchos años de ventaja en el tema.



La popularidad de Java en muchísimos países del mundo se ha generalizado, es parte de pequeños y grandes proyectos en la industria, lo que demuestra su usabilidad a la fecha, y que refleja la importancia de conocer cada una de sus formas más eficientes de programar. Algunas de las empresas que han usado los anteriores lenguajes de programación para su éxito son las siguientes:

Compañías que hacen/han hecho uso de:

Java

- Amazon
- eBay
- Google
- Netflix
- PayPal
- Walmart
- Oracle
- Bank of America
- Uber
- Spotify
- LinkedIn
- Airbnb

PHP

- Viber
- WhatsApp
- Facebook
- 9GAG
- Salesforce
- The next Web
- Wikipedia
- Slack

Como podemos ver, las compañías más poderosas actualmente hacen uso de lenguajes que estudiaremos y que se adapten a sus necesidades, que sean robustos y que puedan escalar fácilmente.

Java: descripción general

Java tiene la característica de ser un lenguaje de programación y, una plataforma para ejecución de aplicaciones bajo su tecnología.

Java como lenguaje de programación

Es un lenguaje de alto nivel, que tiene algunas características como:

- Simple
- Orientado a objetos
- Distribuido
- Multitarea
- Dinámico
- Robusto, seguro y de alto rendimiento

Describiendo algunas de las anteriores características, Java se basa en escribir programas envueltos en **simplicidad**, para que los programadores puedan producir aplicaciones eficientes, quitando algunas características de los anteriores lenguajes que le proceden como C++, creando una programación moderna. Otro de los objetivos destacados en el diseño del lenguaje es que sea familiar para los programadores, su sintaxis es similar a muchos de los lenguajes orientados a objetos, así que la curva de aprendizaje no debería ser amplia y en poco tiempo se estará produciendo aplicaciones de gran nivel. Como muestra de ello, el famoso programa HolaMundo/HelloWorld, que vemos en la documentación de todos los lenguajes cuando iniciamos a programar, escrito en Java, lo haríamos de la siguiente forma.

```
class HolaMundo {
static public void main(String args[]) {
 System.out.println("¡Hola Mundo!");
}
}
```

Más adelante veremos la explicación detallada de cada uno de los elementos contenidos en las anteriores líneas de código, pero a simple vista se trata de una sintaxis de clase y métodos bastante fácil de entender, que es familiar e intuitivo para la lectura de los programadores.

Otra de sus principales características es que es **Orientado a objetos** y por lo cual se ha seleccionado para poder poner en práctica todos los conceptos que nos permitirán

crear programas ordenados y modernos, un paradigma que se adapta muy bien a los programas cliente servidor y sobretodo ahora en aplicaciones distribuidas.

La capacidad de los objetos es indudable, a diferencia de los procedimientos ordinarios en programación, objetos pueden crearse, pasarse por red, almacenarse en base de datos y reutilizarlos para crear flujos de trabajo más organizados en los equipos de desarrollo. Veremos a detalle cómo hace Java para trabajar con objetos de manera eficiente y ordenada.

Tiene una arquitectura **robusta**, las aplicaciones distribuidas actualmente están teniendo mucho auge, la posibilidad de ejecutar aplicaciones en diferentes ecosistemas, manejando una arquitectura neutral y portátil.

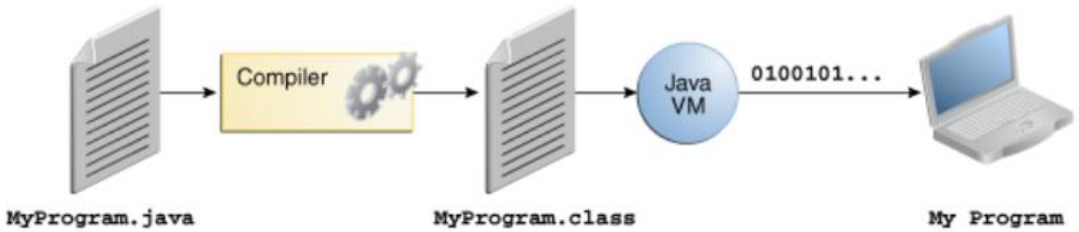
Java es un lenguaje **interpretado** y **dinámico**, con la capacidad de que por medio de un intérprete, se pueda ejecutar las aplicaciones escritas en Java de manera portable, en cualquier plataforma hardware y software. Nos permite una rápida detección de errores en el ciclo de desarrollo.

En la actualidad, desarrollar **aplicaciones seguras** implica parte fundamental del momento en el que se decide que lenguaje de programación utilizar para los proyectos, comercio electrónico, intercambio de dinero, y muchas otras áreas en el mercado actual, obligan al equipo de Java a no confiar en nada, en crear capas de seguridad que impidan la creación de código malicioso.

Es **multihilo**, permite la ejecución de tareas de manera paralela, aprovechando la capacidad de procesamiento de los ordenadores.

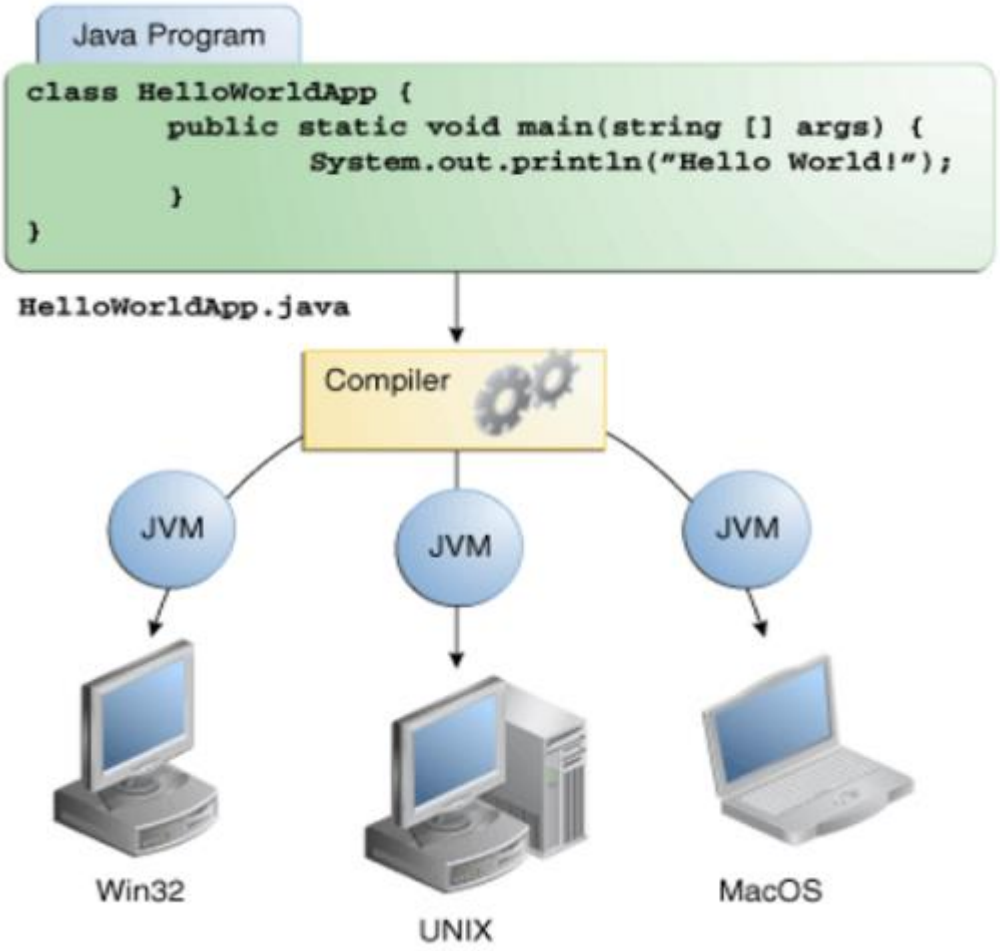
Estructura

Todo el código fuente en java es escrito en extensiones .java, Esos archivos pasan por el compilador *javac* en los archivos .class. En la documentación de Java, el proceso que podemos ver en el ciclo de desarrollo de software es el siguiente:



Se elimina la dependencia de arquitecturas hardware o software, por medio de la JVM (Java Virtual Machine, por sus siglas en inglés), la máquina virtual de java, está

disponible en diferentes sistemas operativos y permite ejecutar aplicaciones a través de ella sin problemas ni fricciones de arquitecturas.



Java: Primeros Pasos

Su sintaxis es familiar a la de otros lenguajes de programación, algunas de las características a destacar son las siguientes:

Tipos de datos

Datos primitivos

Todo en el lenguaje de programación de Java son objetos, incluso los datos primitivos pueden ser trabajados como tal, describimos los siguientes tres grupos:

Datos numéricos

```
byte: 8 bit
short: 16 bit
int: 32 bit
long: 64 bit
float: 32 bit
double: 64 bit
double precio = 23.34;
```

Caracteres

```
char: 16 bit

char micaracter = 'D';
```

Booleanos

```
boolean respuesta = true

boolean respuesta = false
```

Operadores aritméticos:

+	Suma:
-	Resta
*****	Multiplicación:
/	División
%	Mod(Sobrante de la división entre enteros)

Operadores lógicos:

==	Es igual
!=	Es distinto
<, <=, >, >=	Menor, Menor o igual, Mayor, Mayor o igual
&&	Operador And
||	Operador Or
!	Operador Not

Arreglos

Puedes declarar arreglos de cualquier tipo, incluso arreglos de arreglos, para hacerlos multidimensionales.

```
double[] precios = new double[3];
```

Inicialización

```
int[] edades = {
  51,
  32,
  21
 }; //Arreglo que define tres elementos
```

```
String[] apellidos = {
 "Marín",
 "Sarmiento",
 "Ayala"
}; //Arreglo que define tres elementos apellido
```

Agregando elementos al arreglo, el código básico es el siguiente:

```
public class ArregloGrupos {

public static void main(String arg[]) {
   String[] grupo = new String[4];

   grupo[0] = "Nirvana";
   grupo[1] = "Reik";
   grupo[2] = "Guns & Roses";
   grupo[3] = "Aerosmith";
  }
 }
```

Strings

Para declaración de cadenas de caracteres en Java, usamos la siguiente sintaxis:

```
String saludo = "Hola mundo";
```

Para concatenar cadenas de caracteres, usamos la siguiente sintaxis:

```
System.out.println("Hay " + cantidad + " estudiantes en la clase");
```

Donde cantidad es una variable de tipo entero que indica el número de estudiantes que perteneces a la clase.

Impresión de valores en consola

```
public class Program {

    public static void main(String arg[]) {

     System.out.println(“Imprimiendo el tamaño de un arreglo: “+arreglo.length);
     System.out.println(arreglo[0]); //Imprimiendo el valor del arreglo en la posición 0
     System.out.println(arreglo[1]);

     System.out.println("Hola Programadores"); //Imprimiendo una etiqueta de texto

    }
   }
```

Estructura de decisiones

```
if (condición) {
 //instrucciones 1
} else {
 //Si no se cumple la condición, entra a este bloque
 //instrucciones
}
```

Ciclos

Nos permiten ejecutar una o más lineas de código de forma repetitiva, definiendo el número de iteracciones, dependiendo de su tipo, tienen un valor inicial, uno final y el valor incremental entre cada paso del programa:

For

```
for (int i = valor_inicial; i <= valor_final; i = i +
valor_incremento) {
 //Instrucciones y lógica de programa
 //Valor incremento nos indica en que rango aumentaremos el
siguiente ciclo, + 1, por ejemplo, define que    el ciclo
aumentará de 1 en 1
}
```

While

```
boolean prueba = true; //Variable booleana que condiciona las
ejecuciones del ciclo
 while (prueba) { //Mientras la variable prueba tenga el valor
true, se ejecutará el siguiente código
  System.out.println("Esto es un mensaje, lo veras una vez");
  prueba = false; //Asignamos un nuevo valor a la variable
prueba, el ciclo no se volverá a ejecutar
 }
```

Un ejemplo más:

El siguiente programa imprime la cadena de texto "Soy un buen programador" 10 veces

```
public class Cuadrado {
  public static void main(String args[]) {
   //Inicio de la programación para el ciclo
   int contador = 1; //Declarar e inicializar el valor de la
variable contador
   while (contador <= 10) {
    System.out.println("Soy un buen programador\n");
    contador++; //Incrementa la variable contador
   }
  }
 }
```

do-while

Un poco diferente al anterior, aquí realizamos ciertas acciones antes de entrar a la condición del ciclo.

```
public class CuentaDigitosNumero {
 public static void main(String args[]) {
  int número = 455;
  int dígitos = 0;
  do { //Realizamos las siguientes acciones antes de entrar al
ciclo
   número /= 10;
   dígitos++; //Hasta que número sea menos que 0, dejará de
hacer este bloque
  }
  while (número > 0);
  System.out.println(dígitos);
 }
}
```

Ejercicios prácticos # 1: Programando en Java

Realizar los siguientes ejercicios básicos para imprimir en consola los datos de los siguientes requerimientos:

1. Crear el código que imprima la siguiente salida: "Hola, soy Arnol Gutiérrez" (Usar su nombre)
2. Crear un programa que imprima en consola todas las operaciones aritméticas de dos números enteros (suma, resta, multiplicación, división, mod)
3. Dadas las variables de tipo int M = 6, T = 1, K = -10 indicar si la evaluación de estas expresiones daría como resultado verdadero o falso:
    a. M > T
    b. T / K == -5
    c. (M+T == 7) || (M-T == 5)
4. Crear un arreglo que guarde e imprima 10 nombres de tus compañeros de la clase.
5. Crear un arreglo multidimensional que guarde más datos personales tus compañeros de clase (nombre, apellido, carrera, lugarTrabajo), tomando como referencia de la información colocada en el foro Conociéndonos. Llenar 5 registros Ejemplo:

Daniel	Medina	Electronica	TEST
Monica	Jiz	Computacion	IMSA

6. Crear un programa que imprima en consola un nombre de estudiante, una nota y una etiqueta que diga Aprobado o Reprobado, dependiendo del valor de la nota que tenga. Ejemplo:

Daniel	65	Reprobado
Monica	89	Aprobado

7. Escribe un programa que imprima mediante un ciclo los números pares del 2 al 100

1. Ejemplo de aplicación en Java

Hola mundo en Java

Como es normal, cuando iniciamos a codificar en un nuevo lenguaje de programación, es común desarrollar el primer ejercicio imprimiendo un "Hola mundo", es el comienzo de una curva de aprendizaje que se vuelve interesante cuando empezamos a conocer más del lenguaje.

La estructura del código para imprimir un Hola mundo es la siguiente:

```
/**
 * Aplicación HolaMundo para imprimir en consola
 * el mensaje "¡Hola mundo!"
 */
public class HolaMundo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("¡Hola mundo!"); // Mostrar el texto.
    }
}
```

Estructura

Comentarios

Los comentarios ayudan a que el código de nuestras aplicaciones sea más entendible para nosotros o para otros programadores, podemos definir comentarios multilinea de la siguiente manera:

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
```

O también podemos agregarlos de esta forma:

```
//este es un comentario de una linea
```

y la otra forma de agregarlos consiste en definir los siguientes caracteres:

```
/* Comentario */
```

Clase

La palabra reservada **class** nos ayuda a definir una nueva clase para nuestros programas, seguido de un nombre que identifique la funcionalidad de la misma. Se define entre dos llaves, una de apertura y otra de cierre, más adelante veremos más a detalle todas las características y funcionalidades de las clases.

class HolaMundo { public static void main(String[] args) {

System.out.println("Hello World!"); // Display the string. } }

El método main

Cada aplicación en Java debe contener un método principal cuya forma de declaración está resaltada en negrita en el siguiente bloque de código:

class HelloWorldApp { **public static void main(String[] args) {**

System.out.println("Hello World!"); //Display the string. **}** }

Podemos nombrar el argumento a nuestro gusto, en el ejemplo de arriba está puesto como **args** pero podemos asignarlo como arguments, argh, etc.

El modificador de acceso está colocado en el siguiente orden **public static** por convención, aunque también puede estar colocado en el orden contrario.

```
System.out.println("¡Hola mundo!");
```

Con la línea anterior lo que hacemos es usar la clase System del core de librerías de Java para imprimir una cadena de texto como mensaje de salida.

2. Ejemplo de aplicación en Java

Operaciones aritméticas

En el siguiente ejemplo, crearemos la estructura básica de un programa en Java para sumar, restar, multiplicar y dividir dos números de tipo `double`.

El nombre de la clase es `suma`.

Dos variables creadas e inicializadas con valores para pruebas: `numero1` y `numero2`.

Las 4 variables que guardaran los resultados de las operaciones: `suma`, `resta`, `multiplicación` y `división`.

Los operadores `+`, `-`, `*` y `/`, se utilizan para realizar las anteriores operaciones respectivamente.

```
public class Suma {
    public static void main(String[] args) {

    //Declaración e inicialización de 2 variables enteras
         double numero1 = 10;
         double numero2 = 20;
         double suma, resta, multiplicacion = 0;
         double division = 0;

         suma = numero1 + numero2;
         resta = numero1 - numero2;
         multiplicacion = numero1 * numero2;

    //Controlamos que la división se haga sobre un denominador 0
         if (numero2 != 0){
                division = numero1 / numero2;
         }

    // Imprimimos en consola los resultados
 System.out.println("La suma es: " + suma);
 System.out.println("La resta es: " + resta);
 System.out.println("La multiplicación es: " + multiplicacion);
 System.out.println("La división es: " + division);

   }
}
```

La declaración de variables para el manejo de datos con punto flotante se ha realizado con `double`:

```
double numero1 = 10;
double numero2 = 20;
double suma, resta, multiplicacion = 0;
double division = 0;
```

La impresión en consola se realiza con la siguiente sintaxis:

```
System.out.println("La suma es: " + suma);
```

La concatenación de una cadena de texto y una variable o de dos varibles se utiliza el operador `+`.

La condicionante mostrada a continuación valida que no podamos dividir dos números cuyo denominador sea cero.

```
if (numero2 != 0){
division = numero1 / numero2;
}
```

Repasar este ejercicio en el módulo de Desarrollo de la plataforma.

Recursos

Para Java y PHP trabajaremos en ejercicios como práctica en un módulo desarrollado en la plataforma, en donde podrán probar su lógica y desarrollarse como programadores. El nombre del módulo es:

Desarrollo, lo encontrarán en sus opciones principales del curso.

Y la interfaz principal al entrar a cada una de las secciones de práctica es:

Al lado izquierdo podrán leer una documentación acerca del ejercicio, en el centro el editor con corrector léxico y sintáctico y al final el resultado de la ejecución del código de nuestros programas.

Recursos de software para Java

Podrán elegir trabajar con uno de los dos siguientes:

1. Netbeans

IDE de código libre para trabajar con Java. Será el IDE por defecto para la clase.
Web oficial:
https://netbeans.org/

Instrucciones de instalación
https://netbeans.org/community/releases/82/install.html

2. Intelli IDEA

IDE potente y amigable para trabajar con Java, el más usado en el campo laboral. Los que se sientan cómodos pueden descargar y trabajar con este IDE. Recomendación personal.

Es un software de pago, pero estamos tramitando una licencia con el correo estudiantil de UTH para que puedan usarlo (Les comunicaré al tenerla)

Web Oficial:
https://www.jetbrains.com/idea/

Para PHP

1. Sublime Text

Editor de texto y editor de código fuente, tiene muchísimas características y un rendimiento muy bueno. Será muy útil para trabajar en desarrollo con PHP.

Web oficial para descarga:
https://www.sublimetext.com/

2. PHPStorm

Es un IDE de PHP potente, ideal para trabajar con todas las características y frameworks del lenguaje.

Es un software de pago, pero estamos tramitando una licencia con el correo estudiantil de UTH para que puedan usarlo (Les comunicaré al tenerla)

Web Oficial:
https://www.jetbrains.com/phpstorm/

3. WampServer

Para ejecutar las aplicaciones PHP de forma local en nuestra computadora es necesario un ambiente de desarrollo web, en donde un servidor web HTTP llamado Apache sirve las páginas para el cliente, el código se ejecuta al lado del servidor. Una de las herramientas más usadas en WampServer:
http://www.wampserver.com/en/

Documentaciones oficiales:

Java
http://www.oracle.com/technetwork/java/index-136113.html
PHP
http://php.net/manual/es/

Base de datos de artículos profesionales para trabajos de investigación para el curso:

Google Scholar https://scholar.google.com/
dblp Computer Cience Bibliography http://dblp.uni-trier.de/