PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación**: **Programación de Software**
- **Código del Programa de Formación**: 233104
- **Nombre del Proyecto:** Distema de informacion para la gestión de procesos en el sector productivo (**SIGEPROP**)
- **Fase del Proyecto**: Ejecución
- **Actividad de Proyecto:** Construir la aplicación tecnologica de acuerdo a lo requerido por el cliente.
- **Competencia:** Implementar la solución de software de acuerdo con los requisitos de operación y modelos de referencia.
- **Resultados de Aprendizaje Alcanzar**: Desplegar el software construido de acuerdo con la plataforma tecnológica.
- **Duración de la Guía: 40 horas**

2. PRESENTACIÓN

La importancia de Java radica en la posibilidad de crear o diseñar software y poder ser ejecutado en diferentes plataformas donde se requiere, sin la necesidad de estar modificándolo. Es una herramienta diseñada con muchos elementos parecidos a otros lenguajes pero en una versión mejorada, más sencilla de manejar, con menos complejidad y con barreras de seguridad en su lenguaje y ejecución.



Imagen 1. Logo Java

Fuente: http://www.oracle.com/technetwork/es/java/javase/downloads/index.html

Usar java puede interesarnos si queremos crear programas que se vayan a manejar a través de un interfaz web, programas distribuidos en general, o programas que tengan que funcionar en distintos sistemas sin ningún cambio (programas "portables"), entre otros casos.

Los invitamos, a participar activamente de este proceso de enseñanza aprendizaje a través de cada una de las siguientes actividades pensadas para usted. Bienvenidos.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE.

3.1 Actividades de Reflexión Inicial.

Reconocer la importancia de Java en la informática y en el desarrollo de aplicaciones.

Apreciados aprendices, observe el siguiente video "**Por qué Java es importante**" que se encuentra en la carpeta "Material de Apoyo" o en el siguiente enlace: https://www.youtube.com/watch?v=1l7nDIVReBU y en plenaria en compañía del instructor socialice su punto de vista, aportando en la construcción colaborativa del conocimiento. También puedes observar la presentación en línea "**Lenguaje de programación Java**", disponible en https://es.slideshare.net/Chikito304/lenguaje-de-programacin-java-25615742#:~:text=IMPORTANCIA%20DE%20JAVA%20La%20importancia,la%20necesidad%20de%20estar%20modific%C3%A1ndolo.

- **Duración: 5 horas**
- **Tipo Actividad: Individual-Grupal**
- **Ambiente Requerido:** Ambiente de informática
- **Materiales:** Equipos de cómputo, conexión internet, software ofimático, TV o Proyector.

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.

Actividad No 1. Identificar la importancia de los Fundamentos del Lenguaje Java

El instructor organizara a los aprendices en una **mesa redonda** y mediante una lluvia de ideas darán respuesta a los siguientes interrogantes que permitirán identificar el nivel de conocimiento respecto a la temática de la guía. Pueden consultar el material disponible en la carpeta "Material de Apoyo" o en el siguiente enlace: https://es.slideshare.net/Chikito304/lenguaje-de-programacin-java-25615742#:~:text=IMPORTANCIA%20DE%20JAVA%20La%20importancia,la%20necesidad%20de%20estar%20modific%C3%A1ndolo.

Finalmente, los aprendices socializarán el mapa mental, cuadro, mapa conceptual, etc., construido en cualquier herramienta informática, a partir de las siguientes temáticas o preguntas.

1. Describir las características generales de Java
2. ¿Qué sabes acerca de los términos Objeto, Clase, Paquete, interface, Herencia, Encapsulamiento?
3. ¿Cuáles son las características o atributos principales de un Objeto y de una clase?
4. Mencione las diferencias entre un objeto y una clase.
5. ¿Por qué es importante la herencia en la Programación Orientada a Objetos?
6. Describir las Ventajas de Java.

El instructor orientara la lluvia de ideas y al finalizar retroalimentara las respuestas dadas por los aprendices.

- **Duración: 5** horas.
- **Tipo de actividad:** Individual –Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Materiales:** Marcador, tablero, Equipos de cómputo, conexión internet, software ofimático, TV o Proyector.
- **Material de apoyo.** Video "**Cómo hacer un mapa conceptual en Cmpatools**". Enlace: https://www.youtube.com/watch?v=l6qGTo_8dRc

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

Aplicar conceptos de POO a la construcción de aplicaciones basados en casos de estudio que den solucion a los requerimientos planteados por el cliente.

Java es un lenguaje de programación orientada a objetos (object oriented programming) que se incorporó al ámbito de la informática en los años noventa. Desde entonces, su popularidad ha ido aumentado y hoy en día Java es uno de los lenguajes de programación más utilizados del mundo. En la actualidad puedes descargar todo lo relacionado con java gratuitamente incluyenfo el Java Virtual Machine (jvm) o el Open Java Development Kit (jdk) de "fuente abierta" o "código abierto", desde su sitio oficial.

A continuación, se presentan un conjunto de diferentes actividades que le permitirán desarrollar tanto los conocimentos como las habilidades para realizar de forma acertada la construcción de aplicaciones en Java.

Actividad No 1: Reconocer la Estructura de un Programa Java.

Un programa puede construirse empleando varias clases. En el caso más simple se utilizará una única clase. Esta clase contiene el programa, rutina o método principal: main() y en éste se incluyen las sentencias del programa principal. Estas sentencias se separan entre sí por caracteres de punto y coma. La estructura de un programa simple en Java es la siguiente:

```
public class ClasePrincipal {
public static void main(String[] args) {
sentencia_1;
sentencia_2;
// ...
sentencia_N;
}
}
```

Como primer ejemplo sencillo de programa escrito en Java se va a utilizar uno que muestra un mensaje por la pantalla del ordenador. Por ejemplo, el programa `Hola.java`:

```
/**
* La clase hola construye un programa que
* muestra un mensaje en pantalla
*/
// Este es un comentario estilo C++
public class Hola {
public static void main(String[] args) {
System.out.println("Hola, ");
System.out.println("me llamo Luis");
System.out.println("…Hasta luego");
}
}
```

Como se ha indicado anteriormente, en un programa de Java todo se organiza dentro de las clases. En el ejemplo anterior, **Hola es el nombre de la clase principal y del archivo que contiene el código fuente**. Todos los programas o aplicaciones independientes escritas en Java tienen un método main o principal que, a su vez, contiene un conjunto de sentencias. En Java los conjuntos o bloques de sentencias se indican entre llaves { }. En el caso anterior, el conjunto de sentencias se reduce a tres sentencias, que son llamadas a dos métodos predefinidos en Java (print y println) que permiten visualizar texto por el dispositivo de salida de datos por defecto (la pantalla).

En equipos de trabajo de 3 aprendices, implemente por lo menos **cinco (5) programas Java** desarrollados durante el trimestre anterior, utilizando el entorno de desarrollo **IDE NETBEANS.** Una vez terminados los ejercicios y con la participación de todos los aprendices, se deberá ***socializar*** los cinco resultados o programas en Java, y se realizará la retroalimentación pertinente por parte del instructor y se despejaran las dudas que surjan sobre la temática.

Además, en los mismos equipos de trabajo de 3 aprendices, después de realizar la consulta y lectura respectiva relacionada con el tema: ***LAS PALABRAS RESERVADAS EN JAVA***, deben realizar una **Infografía**, la cual se socializará ante los demás compañeros y el instructor.

Para llevar a cabo el desarrollo de la actividad de aprendizaje, se recomienda el material de apoyo dispobible en la **biblioteca virtual del SENA (biblioteca.sena.edu.co**). A continuación, se presentan algunos recursos bibliográficos relacionados con Java que puedes consultar (Puedes seleccionar el libro electrónico que mejor te parezca para realizar la consulta):

• **Thierry RICHARD - Thierry GROUSSARD**. **Los fundamentos del lenguaje Java**. Disponibible en: https://www-eni-training-com.bdigital.sena.edu.co/portal/client/filter/home

• Tutorial Interactivo Multimedia de Fundamentos de Java. Disponible en: http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm

Además, en la carpeta **Material de Apoyo** adjunto a esta guía, encontrará los videos sobre las Infografías o podrán consultarlos directamente en los siguientes enlaces:

- **Infografías:**

Video 1: Qué es una Infografía

https://www.youtube.com/watch?v=lkxd0Lz6x2w

_____(Montenegroagencia, ¿Qué es, para qué y cómo hacer una infografía?, 2015)

Video 2: Tipos de infografía

https://www.youtube.com/watch?v=zSjoPj4YE3c

_____(Yoututorial, Tipos de infografía, 2017)

Video 3: Cómo hacer una infografía de forma sencilla
https://www.youtube.com/watch?v=MKadmIqWUeM

Cuentitis Aguda. Como realizar una infografía de forma sencilla, 2019

https://www.youtube.com/watch?v=lkxd0Lz6x2w

_____(Montenegroagencia, ¿Qué es, para qué y cómo hacer una infografía?, 2015)

- **Duración: 5** horas.
- **Tipo de actividad:** Individual –Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Materiales:** Marcador, tablero, Equipos de cómputo, conexión internet, software ofimático, TV o Proyector.

- **Evidencia de aprendizaje**: Entrega y Socialización de los cinco (5) programas en Java y la Infografía.

Actividad No 2: Sentencias de control del flujo de un programa

Cuando se escribe un programa, se introduce la secuencia de sentencias dentro de un archivo. Sin sentencias de control del flujo, el intérprete ejecuta las sentencias conforme aparecen en el programa de principio a fin. Las sentencias de control de flujo se emplean en los programas para ejecutar sentencias condicionalmente, repetir un conjunto de sentencias o, en general, cambiar el flujo secuencial de ejecución. Las sentencias selectivas o condicionales se verán en este capítulo y las sentencias repetitivas en el siguiente.

- ***Sentencia if-else***

Es una bifurcación o sentencia condicional de una o dos ramas. La sentencia de control evalúa la condición lógica o booleana. Si esta condición es cierta entonces se ejecuta la sentencia o sentencias (1) que se encuentra a continuacion. En caso contrario, se ejecuta la sentencia (2) que sigue a else (si ésta existe). La sentencia puede constar opcionalmente de una o dos ramas con sus correspondientes sentencias.

Sintaxis:

```
if (expresionLogica) {
sentencia_1;
}
```

o bien (con dos ramas):

```
if (expresionLogica) {
sentencia_1;
}
else {
sentencia_2;
}
```

La ***expresionLogica*** debe ir entre paréntesis. Las llaves sólo son obligatorias si las sentencias (1) ó (2) son compuestas (las llaves sirven para agrupar varias sentencias simples). La parte else y la sentencia posterior entre llaves no son obligatorias. En este caso quedaría una sentencia selectiva con una rama como se aprecia en la parte izquierda de la siguiente figura.

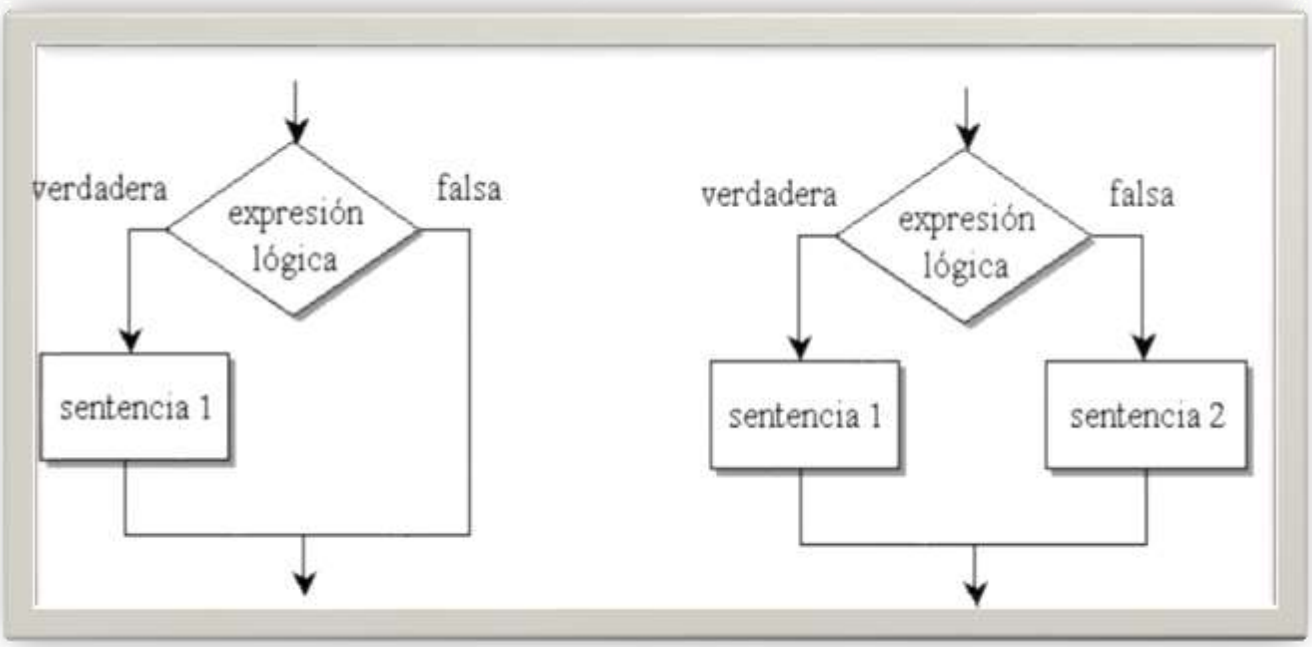


Figura 2. Sentencias de control de flujo if…else…
Fuente: Elaboración propia

Un ejemplo muy sencillo que muestra este tipo de sentencia es el siguiente:

```
// Codigo autoexplicativo
if (a>b) {
System.out.println(”a es mayor que b");
}
else {
System.out.println(”a no es mayor que b");
}
```

Ejemplo de programa completo:

```
/**
* cuadrado: Ejemplo de sentencia if
* Muestra el cuadrado de un valor entero introducido como parametro
* Recopiló Ing. Luis Armando Amaya Q.
*/
public class Cuadrado {
public static void main (String [] args) {
int valor;
if (args.length == 0)
System.out.println("Por favor, introduce un argumento entero");
else {
valor = Integer.parseInt(args[0]);
System.out.println("El cuadrado es " + valor*valor);
}
System.out.println("Hasta pronto, espero que hayas aprendido mucho ");
}
```

}

- **Las sentencias if–else pueden ir <u>anidadas</u> unas dentro de otras en el código fuente del programa. Por ejemplo:**

```
/**
* Ejemplo de sentencias if anidadas
* Indica si el valor entero introducido como parametro es par o no
* Recopiló Ing. Luis Armando Amaya Q.
*/
public class EsPar {
public static void main (String [] args) {
int valor;
if (args.length == 0)
System.out.println("Por favor, introduce un argumento entero");
else {
valor = Integer.parseInt(args[0]);
if (valor % 2 == 0) {
System.out.println("El valor " + valor + " es par");
}
else {
System.out.println("El valor " + valor + " es impar");
}
}
System.out.println("Hasta pronto, espero que hayas aprendido mucho");
}
}
```

- ***<u>Sentencia switch</u>***

Es una sentencia condicional multiramificada o de selección multiple: dependiendo del valor de una variable o expresion entera permite ejecutar una o varias sentencias de entre muchas. La expresión puede ser de un tipo ordinal (de tipo entero byte, short ó int o de tipo carácter char) pero no puede ser de un tipo real o de un tipo cadena.

<u>Sintaxis</u>:

```
switch (expresion) {
case valor_1: sentencias_1; break;
case valor_2: sentencias_2; break;
...
case valor_n: sentencias_n; break;
[default: sentencias_x;]
}
```

Cada sentencia case contiene un único valor distinto del de las demás sentencias case. A continuación del valor se introduce la sentencia o sentencias que se ejecutan en el caso de que el valor indicado coincida con el de la variable o expresión selector. Las sentencias que siguen a cada uno de los valores no se engloban entre llaves, pero suelen ir seguidas de un break. Si la expresión no coincide con ningún valor se ejecuta la sentencia que sigue a default, aunque esta parte (default) no es obligatoria. Si no existe algún break, continua la ejecución de la siguiente opción hasta el siguiente break o hasta el final de la sentencia switch. Lo anterior se representa en la siguiente figura:

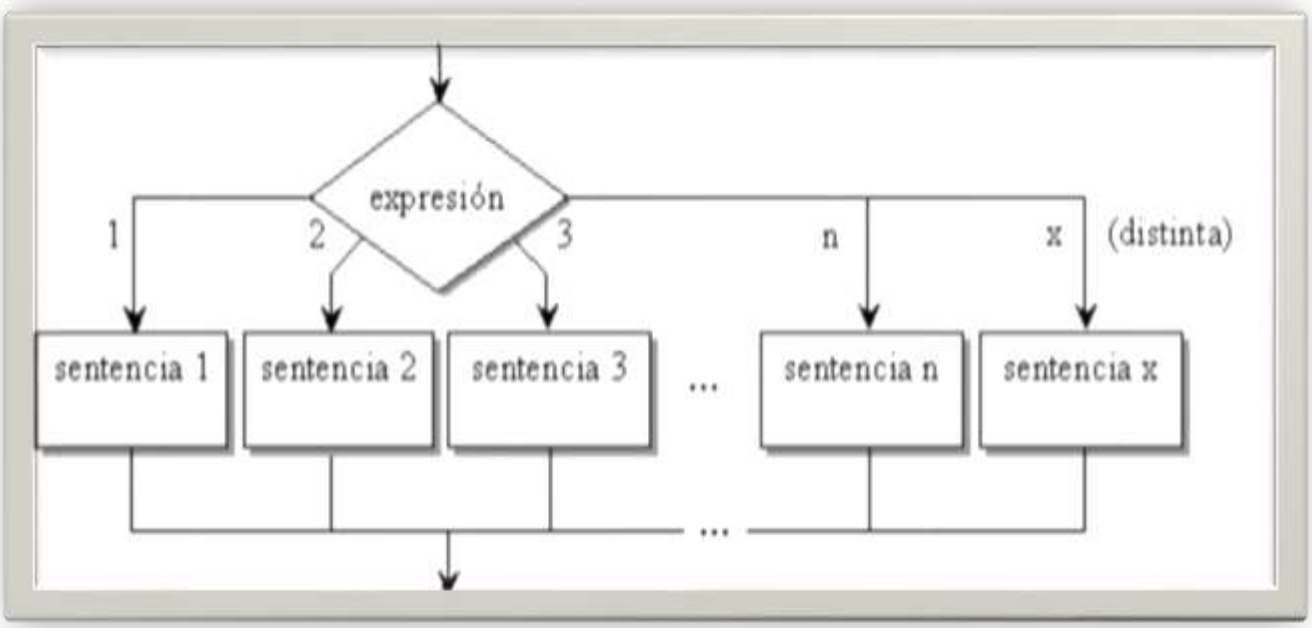


Figura 2. Sentencias de control de flujo if…else…
Fuente: Elaboración propia

A continuación se muestra un ejemplo de programa que emplea la sentencia switch para visualizar en pantalla mensajes distintos, en función del primer caracter del primer argumento introducido en la línea de comandos de la ejecución:

```
/**
* Ejemplo de uso de la sentencia switch
* Recopiló: Ing. Luis Armando Amaya Q.
*/
public class Caracter {
public static void main (String [] args) {
char c;
c=args[0].charAt(0);
switch (c) {
case 'a': System.out.println("Es la vocal a"); break;
case 'e': System.out.println("Es la vocal e"); break;
case 'i': System.out.println("Es la vocal i"); break;
case 'o': System.out.println("Es la vocal o"); break;
case 'u': System.out.println("Es la vocal u"); break;
case 'A': System.out.println("Es la vocal A");
case 'E':
case 'I':
case 'O':
case 'U': System.out.println("Vocal mayuscula"); break;
default: System.out.println("No es una vocal"); break;
}
System.out.println("Hasta pronto, espero que hayas aprendido mucho…");
```

}
}

En equipos de trabajo de 3 aprendices, implemente por lo menos **cinco (3) programas Java** que utilice las estructuras **if….else…..switch**, utilizando el entorno de desarrollo **IDE NETBEANS.** Una vez terminados los ejercicios y con la participación de todos los aprendices, se deberá ***socializar*** los tes resultados o programas en Java, y se realizará la retroalimentación pertinente por parte del instructor y se despejaran las dudas que surjan sobre la temática.

Para llevar a cabo el desarrollo de la actividad de aprendizaje, se recomienda el material de apoyo dispobible en la **biblioteca virtual del SENA (biblioteca.sena.edu.co**). A continuación, se presentan algunos recursos bibliográficos relacionados con Java que puedes consultar (Puedes seleccionar el libro electrónico que mejor te parezca para realizar la consulta):

• **Thierry RICHARD - Thierry GROUSSARD**. **Los fundamentos del lenguaje Java**. Disponibible en: https://www-eni-training-com.bdigital.sena.edu.co/portal/client/filter/home

• Tutorial Interactivo Multimedia de Fundamentos de Java. Disponible en: http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm

- **Duración: 5** horas.
- **Tipo de actividad:** Individual –Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Materiales:** Marcador, tablero, Equipos de cómputo, Internet, software ofimático, TV o Proyector.
- **Material de apoyo. Video: "**Ejercicios con condicionales en Java", disponible en: https://www.youtube.com/watch?v=gL4ZQi6MEic
- **Evidencia de aprendizaje**: Entrega y Socialización de los tres (3) programas implementados en Java

Actividad No. 3. Aplicar técnicas de programación orientada a objetos en Java

A continuación, en el recurso de aprendizaje o guía **"PROGRAMACIÓN ORIENTADA A OBJETOS APLICADA EN JAVA**" del SENA, se presentan un conjunto de diferentes actividades prácticas en el lenguaje Java, que le permitirán desarrollar tanto los conocimentos como las habilidades para la construcción de la aplicación en Java.

En equipos de trabajo de 3 aprendices, con el acompañamiento del instructor, después de realizar el estudio del anterior recurso de aprendizaje realice cada una de las actividades allí planeadas.

El recurso de aprendizaje o guía **"PROGRAMACIÓN ORIENTADA A OBJETOS APLICADA EN JAVA**" del SENA, para la realización de las actividades son directrices generales, sin embargo, el instructor puede emplear esta guía o cualquier otro recurso que le permita complementar, actualizar y fortalecer el proceso de enseñanza y aprendizaje.

También puede consultar otros sitios web, los cuales contienen información pertinente sobre los temas a tratar, como por ejemplo la guía "**Programación Orientada a Objetos**", disponible en http://aprendeenlinea.udea.edu.co/lms/moodle/file.php/561/Poo_-_Fundamentos/presentacion_Java_1.pdf

Además, se recomienda el material de apoyo dispobible en la **biblioteca virtual del SENA (biblioteca.sena.edu.co**). A continuación, se presentan algunos recursos bibliográficos relacionados con Java que puedes consultar (Puedes seleccionar el libro electrónico que mejor te parezca para realizar la consulta):

• **Thierry RICHARD - Thierry GROUSSARD**. **Los fundamentos del lenguaje Java**. Disponibible en: https://www-eni-training-com.bdigital.sena.edu.co/portal/client/filter/home

• **Tutorial Interactivo Multimedia de Fundamentos de Java**. Disponible en: http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm

- **Duración**: 5 horas
- **Tipo de actividad:** Individual – Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Evidencia de aprendizaje**: Socialización y entrega de los ejercicios planteados en la guía
- **Materiales:** Marcador, tablero, Equipos de cómputo, Internet, software ofimático, TV o Proyector.
- **Material de apoyo. Video "Curso Java**", disponible en: https://www.youtube.com/watch?v=XmUz5WJmJVU

Actividad No. 4. Desarrollo de aplicaciones en Java

A continuación, en el recurso de aprendizaje o guía **"DESARROLLO DE APLICACIONES EN JAVA**" del SENA, se presentan un conjunto de diferentes actividades prácticas en el lenguaje Java, que le permitirán desarrollar tanto los conocimentos como las habilidades para la construcción de la aplicación en Java.

En equipos de trabajo de 3 aprendices, con el acompañamiento del instructor, después de realizar el estudio del anterior recurso de aprendizaje realice cada una de las actividades allí planeadas.

El recurso de aprendizaje o guía **"DESARROLLO DE APLICACIONES EN JAVA**" del SENA, para la realización de las actividades son directrices generales, sin embargo, el instructor puede emplear esta guía o cualquier otro recurso que le permita complementar, actualizar y fortalecer el proceso de enseñanza y aprendizaje.

También puede consultar otros sitios web, los cuales contienen información pertinente sobre los temas a tratar, como por ejemplo la guía "**Desarrollo de aplicaciones Java**", disponible en https://www.lawebdelprogramador.com/pdf/145f77e4df92965583030d34d34ea746/22464/descargar.php

Además, se recomienda el material de apoyo dispobible en la **biblioteca virtual del SENA (biblioteca.sena.edu.co**). A continuación, se presentan algunos recursos bibliográficos relacionados con Java que puedes consultar (Puedes seleccionar el libro electrónico que mejor te parezca para realizar la consulta):

• **Thierry RICHARD - Thierry GROUSSARD**. **Los fundamentos del lenguaje Java**. Disponibible en: https://www-eni-training-com.bdigital.sena.edu.co/portal/client/filter/home

• **Tutorial Interactivo Multimedia de Fundamentos de Java**. Disponible en: http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm

- **Duración**: 5 horas
- **Tipo de actividad:** Individual – Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Evidencia de aprendizaje**: Socialización y entrega de los ejercicios planteados en la guía
- **Materiales:** Marcador, tablero, Equipos de cómputo, Internet, software ofimático, TV o Proyector.
- **Material de apoyo. Video "Curso Java"**, disponible en: https://www.youtube.com/watch?v=XmUz5WJmJVU

3.4 Actividades de Transferencia del Conocimiento.

Actividad No 1. Aplicar las técnicas y procedimientos de la programación orientada a objetos en Java para realizar CRUD en una aplicación o prototipo.

Conformar grupos de tres aprendices, y mediante la orientación, asesoría y acompañamiento del Instructor, descargar el material de apoyo para las actividades de transferencia de conocimientos.

Si desea, puede consultar otros sitios web, los cuales contienen información pertinente sobre los temas a tratar, como por ejemplo la guía "**Desarrollo de aplicaciones Java**", disponible en https://www.lawebdelprogramador.com/pdf/145f77e4df92965583030d34d34ea746/22464/descargar.php

- **Duración**: 8 horas
- **Tipo de actividad:** Individual – Grupal
- **Ambiente Requerido:** Ambiente de informática
- **Evidencia de aprendizaje**: Socialización y entrega de la aplicación en Java implementando CRUD
- **Materiales:** Marcador, tablero, Equipos de cómputo, Internet, software ofimático, TV o Proyector.
- **Material de apoyo. Video "Curso Java"**, disponible en: https://www.youtube.com/watch?v=XmUz5WJmJVU

4. ACTIVIDADES DE EVALUACIÓN

Estimado aprendiz el proceso de Evaluación de las competencias, es decir de los conocimientos, habilidades y destrezas alcanzadas por usted se ha realizado en cada una las actividades realizados durante toda la formación mediante diferentes técnicas e instrumentos de evaluación pero de igual manera se orientara un proceso cuantitativo de la formación solicitándole que ingrese a la plataforma LMS y en el espacio del curso, denominado Foros de evaluación, participe resolviendo la pregunta generadora planteada por el instructor, para valoración del proceso formativo y de las actividades desarrolladas durante el curso.

- **Duración de la actividad:** 2 Horas.
- **Tipo de actividad:** Individual – Grupal
- **Ambiente Requerido** : Sala de computo
- **Materiales** : Equipos de computo, internet, Software ofimático
- **Evidencia de aprendizaje:** Participación foros de Evaluacion propuestos en la plataforma LMS.

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Respuesta a preguntas mediante evaluación de conocimientos talleres teórico-prácticos. Dominio de los lenguajes de programación JAVA, comandos, sintaxis. **Evidencias de Desempeño:** Resolución de problemas mediante ejercicios prácticos en Java y herramientas tecnológicas Laboratorios prácticos utilizando el Software IDE NETBEANS Realización de Mapa conceptual; cuadro, infografía. **Evidencias de Producto:** Ejercicios resueltos en archivos tipo Java. Infografías, Presentaciones en power point o en cualquier otra herramienta informática.	Utiliza software y herramientas informáticas para realizar aplicaciones en Java Construye la interfaz del aplicativo, siguiendo los parámetros establecidos en el diseño. Utiliza las herramientas de desarrollo proporcionadas por el **IDE NETBEANS**, para la codificación de los módulos del sistema, aplicando las funciones propias del lenguaje de programación **JAVA**. Desplegar el software construido de acuerdo con la plataforma tecnológica instala el software para servidor de aplicaciones de acuerdo con el procedimiento establecido. instala el software para servidor de base de datos de acuerdo con el procedimiento establecido. instala el software codificado de acuerdo con procedimientos establecidos	**Técnicas**: Formulación de preguntas **Instrumento**: Cuestionario **Técnicas**: Observación, socialización, Observación, elaboración y valoración de producto. **Instrumentos**: Lista de chequeo

Prototipo de la aplicación en Java, interfaces y módulos **CRUD** del sistema de información, pruebas de software.		

5. GLOSARIO DE TÉRMINOS

Abstracción: Propiedad y/o técnica de software que oculta los detalles de la implementación. Java soporta abstracción de clases y abstracción de métodos. La abstracción de métodos se define separando el uso de un método sin conocer como está implementado ese método. Si decide combinar la implementación, el programa cliente será afectado. De modo similar la abstracción de clases oculta la implementación de la clase del cliente.

Atributo: Los atributos de una clase sirven, en principio, para almacenar datos de los objetos que se instancian a partir de esa clase

Clase: Colección encapsulada de datos y operaciones que actúan sobre los datos. El concepto de clase es fundamental en programación orientada a objetos. Una clase consta de métodos y datos. Los métodos de una clase definen el conjunto de operaciones permitidas sobre los datos de una clase (sus atributos). Una clase puede tener muchas instancias de la clase u objetos.

Constructor: Método especial utilizado para inicializar el estado de un nuevo objeto. El constructor permite crear objetos utilizando el operador new. El constructor tiene exactamente el mismo nombre que la clase que lo contiene. Los constructores se pueden sobrecargar con el objetivo de facilitar la construcción de objetos con diferentes tipos de valores iniciales.

Herencia: Una relación entre clases en que una subclase se extiende desde una superclase.

IDE: Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrateddevelopmentenvironment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

JSF: JavaServer Faces. tecnología de la plataforma Java Platform EE que soporta la

creación de páginas web dinámicas.

JPA: acrónimo de Java Persistency API. Es la capa de persistencia que contiene las

clases que interactúan directamente con el modelo o base de datos.

JSR: Java Specification Request. Solicitud de cambios a las especificaciones Java.

Método de la clase: Sinónimo de método estático. Un método que se puede invocar sin crear una instancia de la clase. Para definir métodos de clases, se ha de poner un modificador static en la declaración del método.

MVC: Modelo Vista Controlador. Patrón de diseño comúnmente usado en el desarrollo web.

Netbeans: Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo.

Objeto: Un objeto es una representación de un valor del tipo implementado por su clase. La clase declara un objeto de variables, instancia que forman la estructura de un objeto y un conjunto de métodos que se pueden llamar en un objeto.

Polimorfismo: el polimorfismo se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía.

Software: Se conoce como software al soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware. La interacción entre el software y el hardware hace operativo un ordenador (u otro dispositivo), es decir, el Software envía instrucciones que el Hardware ejecuta, haciendo posible su funcionamiento.

Variable: Es una característica que puede fluctuar y cuya variación es susceptible de adoptar diferentes valores, los cuales pueden medirse u observarse. Las variables adquieren valor cuando se relacionan con otras variables, es decir, si forman parte de una hipótesis o de una teoría. En este caso se las denomina constructos o construcciones hipotéticas.

6. REFERENTES BLIBIOGRÁFICOS

- B. Selic, G. Gullekson, J. McGee, and I. Engelberg. (2006). "ROOM: An Object-OrientedMethodology for Developing Real-Time Systems". In Fifth International Workshop on Computer-Aided Software Engineering (CASE'92), Montreal,Canada.
- Biblioteca SENA http://biblioteca.sena.edu.co/
- Calle, Á. S., Bellas, P. G., Abril, A. P., & Serrano, J. V. (2012). Diseñar y programar, todo es empezar: Una introducción a la programación orientada a objetos usando UML y Java. Dykinson. Disponible en http://www.digitaliapublishing.com.bdigital.sena.edu.co/visor/13684
- Desarrollo de aplicaciones en Java. Recuperado de : http://www.robertobandini.it/studio/progetti-svolti/elaborato-di-sistemi-operativi-chat-java/
- Fernández, H. F. Programación orientada a objetos con Java. Ecoe Ediciones. http://www.digitaliapublishing.com.bdigital.sena.edu.co/visor/30030
- Guía de Java. Recuperado de: http://www.codeproject.com/Articles/26675/Beginner-s-Guide-to-Threading-in-NET-Part-of-n
- Geary, D. & Horstmann, C. (2010). Core JavaServer Faces. Madrid: Prentice-Hall.
- General Java Development Learning Trail. (2017). Recuperado de https://netbeans.org/kb/docs/java/quickstart.html
- Oracle (2013). Java 2 Platform Standard Edition 5.0. API Specification. Recueprado de http://docs.oracle.com/javase/6/docs/api/

- Schilt, H (2011). "Java 2.Manual de referencia". McGraw-Hill.
- **Thierry RICHARD - Thierry GROUSSARD**. **Los fundamentos del lenguaje Java**. Disponibible en: https://www-eni-training-com.bdigital.sena.edu.co/portal/client/filter/home
- **Tutorial Interactivo Multimedia de Fundamentos de Java**. Disponible en: http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm
- Wu., T.C (2006). "Introducción a la programación orientada a objetos". McGraw-Hill.

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Luis Armando Amaya Q.**	**Instructor**	**CEAI**	**07 de Febrero de 2021**

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					