Instalación de Java en Windows

Guía paso a paso

Paso 1: Descargar Java Development Kit (JDK)

- Visita la página oficial de Oracle: oracle.com/java/technologies/downloads
- Selecciona la versión de JDK para Windows
- Haz clic en el enlace de descarga para el instalador (.exe)

Paso 2: Iniciar la instalación

- Localiza el archivo descargado (ej. jdk-XX.X.X_windows-x64_bin.exe)
- Haz doble clic para iniciar el instalador
- Sigue las instrucciones del asistente de instalación

Paso 3: Configurar las variables de entorno

1. Busca "Variables de entorno" en el menú de inicio
2. Haz clic en "Editar las variables de entorno del sistema"
3. En la pestaña "Avanzado", selecciona "Variables de entorno"

Paso 4: Configurar JAVA_HOME

1. En "Variables del sistema", haz clic en "Nueva"
2. Nombre de la variable: JAVA_HOME
3. Valor: Ruta de instalación de Java (ej. C:\Program Files\Java\jdk-XX.X.X)

Paso 5: Actualizar la variable PATH

1. En "Variables del sistema", busca la variable "Path"
2. Selecciónala y haz clic en "Editar"
3. Haz clic en "Nuevo" y añade: %JAVA_HOME%\bin

Paso 6: Verificar la instalación

1. Abre la línea de comandos (cmd)
2. Escribe los siguientes comandos:

```
java –version
javac –version
```

3. Deberías ver la versión de Java instalada

¡Listo para programar en Java!

- Java Development Kit (JDK) instalado
- Variables de entorno configuradas
- Entorno preparado para desarrollar aplicaciones Java

Instalación de Java Extension Pack en VS Code

Guía paso a paso

Paso 1: Instalar Visual Studio Code

- Descarga VS Code desde code.visualstudio.com
- Ejecuta el instalador y sigue las instrucciones
- Abre VS Code una vez instalado

Paso 2: Acceder a las extensiones

- Haz clic en el icono de extensiones en la barra lateral izquierda
- O usa el atajo: `Ctrl+Shift+X` (Windows/Linux) o `Cmd+Shift+X` (Mac)

Paso 3: Buscar Java Extension Pack

- En la barra de búsqueda, escribe "Java Extension Pack"
- Localiza el pack oficial de Microsoft

Paso 4: Instalar Java Extension Pack

- Haz clic en el botón "Install"
- Espera a que se complete la instalación
- VS Code puede solicitar reiniciar

Paso 5: Verificar la instalación

- Abre la paleta de comandos: `Ctrl+Shift+P` (Windows/Linux) o `Cmd+Shift+P` (Mac)
- Escribe "Java: Overview"
- Selecciona para ver el resumen de características de Java

Paso 6: Configurar el JDK

- VS Code detectará automáticamente el JDK instalado
- Si no, ve a File > Preferences > Settings
- Busca "java.home" y establece la ruta al JDK

Paso 7: Crear un proyecto Java

- Abre la paleta de comandos
- Escribe "Java: Create Java Project"
- Sigue el asistente para crear un nuevo proyecto

Paso 8: Explorar características

- IntelliSense y autocompletado
- Depuración de código Java
- Ejecución de tests
- Administración de dependencias con Maven/Gradle

¡Listo para desarrollar en Java con VS Code!

- Java Extension Pack instalado
- Entorno configurado
- Herramientas de desarrollo Java disponibles

El Método Main en Java

El punto de entrada de toda aplicación Java

¿Qué es el método main?

- Es el punto de entrada de una aplicación Java
- Cada programa Java debe tener al menos un método main
- Es donde comienza la ejecución del programa

```
public static void main(String[] args) {
    // Código del programa
}
```

Sintaxis del método main

```
public static void main(String[] args)
```

- `public`: Modificador de acceso
- `static`: Palabra clave que indica que es un método de clase
- `void`: Tipo de retorno (no devuelve nada)
- `main`: Nombre del método
- `String[] args`: Parámetro de línea de comandos

¿Por qué public?

- `public`: Permite que la JVM (Java Virtual Machine) acceda al método
- La JVM no es parte de la clase, por lo que necesita acceso público

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

¿Por qué static?

- `static`: Permite que la JVM llame al método sin crear una instancia de la clase
- Los métodos static pertenecen a la clase, no a objetos específicos

```
public class Calculadora {
    public static int sumar(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        System.out.println(Calculadora.sumar(5, 3));
    }
}
```

Parámetro String[] args

- `String[] args`: Array de strings para argumentos de línea de comandos
- Permite pasar información al programa cuando se ejecuta

```
public class Saludar {
    public static void main(String[] args) {
        if (args.length > 0) {
            System.out.println("Hola, " + args[0] + "!");
        } else {
            System.out.println("Hola, Mundo!");
        }
    }
}
```

Ejecución de un programa Java

1. Compilar el código fuente:

```
javac MiPrograma.java
```

2. Ejecutar el programa compilado:

```
java MiPrograma
```

3. Con argumentos:

```
java MiPrograma argumento1 argumento2
```

Variaciones válidas del método main

```
public static void main(String[] args)
public static void main(String args[])
public static void main(String... args)
```

Todas son equivalentes y aceptadas por el compilador de Java.

Buenas prácticas

1. Mantén el método main limpio y conciso
2. Utiliza otros métodos para la lógica del programa
3. Maneja los argumentos de línea de comandos adecuadamente
4. Documenta el uso esperado de los argumentos

```
public static void main(String[] args) {
    if (args.length < 2) {
        System.out.println("Uso: java Programa  ");
        return;
    }
    // Lógica del programa
}
```

¡Ahora estás listo para crear programas Java!

- Entiendes la importancia del método main
- Puedes crear el punto de entrada de tus aplicaciones
- Sabes cómo manejar argumentos de línea de comandos

Tipos de Datos en Java

Fundamentos de la programación en Java

Clasificación de Tipos de Datos en Java

1. Tipos Primitivos
2. Tipos de Referencia

Tipos Primitivos

- Almacenan un solo valor
- Tienen un tamaño fijo en memoria
- Son los bloques de construcción básicos

Categorías:

1. Números enteros
2. Números de punto flotante
3. Caracteres
4. Booleanos

Tipos de Datos Enteros

```
byte b = 100;
short s = 1000;
int i = 100000;
long l = 1000000000L; // Nota la 'L' al final
```

Tipos de Datos de Punto Flotante

```
float f = 3.14f;  // Nota la 'f' al final
double d = 3.14159265359;
```

Tipo de Dato Carácter

- `char`: Representa un solo carácter Unicode
- Tamaño: 16 bits
- Rango: 0 a 65,535

```
char c1 = 'A';
char c2 = '\u0041';  // Representación Unicode de 'A'
```

Tipo de Dato Booleano

- `boolean`: Representa un valor de verdad
- Sólo puede tener dos valores: `true` o `false`

```
boolean isJavaFun = true;
boolean isCodingHard = false;
```

Tipos de Referencia

- Hacen referencia a objetos
- Almacenan la dirección del objeto en memoria
- Incluyen clases, interfaces y arrays

```
String name = "Java";
int[] numbers = {1, 2, 3, 4, 5};
ArrayList list = new ArrayList<>();
```

String: Un tipo especial

- `String` es una clase, pero se comporta como un tipo primitivo
- Es inmutable (no se puede cambiar después de crearse)

```
String greeting = "Hello, World!";
String name = new String("Java");
```

Conversión entre Tipos de Datos

Conversión Implícita (Widening)

```
int myInt = 100;
long myLong = myInt; // Automático
```

Conversión Explícita (Narrowing)

```
double myDouble = 9.78;
int myInt = (int) myDouble; // Casting manual
```

Constantes en Java

- Usa la palabra clave `final` para declarar constantes
- Por convención, se nombran en mayúsculas

```
final double PI = 3.14159;
final int MAX_USERS = 100;
```

Resumen

- Java tiene 8 tipos primitivos: byte, short, int, long, float, double, char, boolean
- Los tipos de referencia incluyen clases, interfaces y arrays
- Entiende las diferencias y elige el tipo adecuado para cada situación

¡Practica con los tipos de datos!

- Experimenta con diferentes tipos
- Observa cómo se comportan en diferentes operaciones
- Recuerda manejar las conversiones con cuidado

Casteo en Java

Convertir un tipo de dato en otro

¿Qué es el Casteo?

- Proceso de convertir un tipo de dato en otro
- Fundamental en Java para manejar diferentes tipos de datos

Tipos de Casteo

1. Conversión Automática (Widening Casting)
2. Casteo Requerido (Narrowing Casting)

Conversión Automática

- De tipo más pequeño a más grande
- Java lo realiza automáticamente
- Sin pérdida de datos

```
byte -> short -> int -> long -> float -> double
```

Ejemplo de Conversión Automática

```java
int myInt = 9;
double myDouble = myInt; // Automático

System.out.println(myInt);      // 9
System.out.println(myDouble);   // 9.0
```

Casteo Requerido

- De tipo más grande a más pequeño
- Debe hacerse manualmente
- Riesgo de pérdida de datos

Sintaxis: `(tipo_destino) valor`

Ejemplo de Casteo Requerido

```
double myDouble = 9.78;
int myInt = (int) myDouble; // Manual

System.out.println(myDouble);   // 9.78
System.out.println(myInt);      // 9
```

Consideraciones Importantes

1. Pérdida de datos en casteo requerido
2. Posible desbordamiento
3. Truncamiento en casteo float/double a entero
4. Casteo entre char e int
5. Promoción automática en expresiones

Ejemplo de Desbordamiento

```
int largeValue = 130;
byte smallByte = (byte) largeValue;
System.out.println(smallByte);  // -126
```

Ejemplo de Casteo char-int

```
char myChar = 'A';
int myInt = (int) myChar;
System.out.println(myInt);  // 65 (ASCII de 'A')
```

Promoción en Expresiones

```java
int a = 5;
double b = 2.0;
double result = a / b;  // 'a' promovido a double
System.out.println(result);  // 2.5
```

Conclusiones

- El casteo es esencial en Java
- Usar con precaución para evitar errores
- Comprender las implicaciones del casteo
- Practicar para dominar este concepto

Operadores de Asignación, Incremento y Decremento en Java

Operadores de Asignación

- Asignan valores a variables
- Operador básico: `=`
- Operadores compuestos: combinan asignación y operación aritmética

Operadores de Asignación Compuesta

- `+=` : Suma y asigna
- `-=` : Resta y asigna
- `*=` : Multiplica y asigna
- `/=` : Divide y asigna
- `%=` : Calcula el módulo y asigna

Ejemplos de Asignación Compuesta

```
int a = 5;
a += 3;  // a es ahora 8
a -= 2;  // a es ahora 6
a *= 2;  // a es ahora 12
a /= 3;  // a es ahora 4
a %= 3;  // a es ahora 1
```

Operadores de Incremento y Decremento

- Incremento: `++`
- Decremento: `--`
- Aumentan o disminuyen el valor en 1

Uso de Incremento y Decremento

- Como prefijo: `++x` o `--x`
- Como sufijo: `x++` o `x--`

```
int x = 5;
x++;    // x es ahora 6
++x;    // x es ahora 7
x--;    // x es ahora 6
--x;    // x es ahora 5
```

Prefijo vs Sufijo

- Prefijo: operación antes de usar el valor
- Sufijo: operación después de usar el valor

Ejemplo: Prefijo vs Sufijo

```
int a = 5, b = 5;
System.out.println(++a);  // Imprime 6
System.out.println(b++);  // Imprime 5
System.out.println(a);    // Imprime 6
System.out.println(b);    // Imprime 6
```

Consideraciones Importantes

1. Asignación compuesta: conversión implícita de tipos
2. Incremento/decremento: solo con variables, no literales
3. Cuidado en expresiones complejas
4. Priorizar legibilidad cuando sea necesario

Conclusiones

- Operadores poderosos para código conciso
- Usar con comprensión de sus implicaciones
- Practicar para dominar su uso efectivo

Tipos de Funciones en Java

1. Funciones sin parámetros y sin retorno

- No reciben input ni devuelven valor
- Ejemplo:

```
public void saludar() {
    System.out.println("¡Hola, mundo!");
}
```

2. Funciones con parámetros y sin retorno

- Reciben input pero no devuelven valor
- Ejemplo:

```
public void saludarPersona(String nombre) {
    System.out.println("¡Hola, " + nombre + "!");
}
```

3. Funciones sin parámetros y con retorno

- No reciben input pero devuelven un valor
- Ejemplo:

```
public int obtenerNumeroAleatorio() {
    return (int) (Math.random() * 100);
}
```

4. Funciones con parámetros y con retorno

- Reciben input y devuelven un valor
- Ejemplo:

```
public int sumar(int a, int b) {
    return a + b;
}
```

5. Funciones recursivas

- Se llaman a sí mismas
- Ejemplo:

```
public int factorial(int n) {
    if (n == 0 || n == 1) return 1;
    return n * factorial(n - 1);
}
```

6. Funciones con número variable de argumentos

- Reciben un número variable de argumentos
- Ejemplo:

```
public int sumarTodos(int... numeros) {
    int suma = 0;
    for (int num : numeros) suma += num;
    return suma;
}
```

7. Funciones anónimas (lambdas)

- Funciones sin nombre
- Ejemplo:

```
Operacion suma = (a, b) -> a + b;
Operacion multiplicacion = (a, b) -> a * b;
```

8. Métodos estáticos

- Pertenecen a la clase, no a instancias
- Ejemplo:

```
public class Matematicas {
    public static double calcularArea(double radio) {
        return Math.PI * radio * radio;
    }
}
```

9. Métodos de instancia

- Pertenecen a instancias específicas de una clase
- Ejemplo:

```
public class Rectangulo {
    private double ancho, alto;

    public double calcularArea() {
        return ancho * alto;
    }
}
```

Conclusiones

- Las funciones son versátiles y fundamentales en Java
- Cada tipo tiene sus usos específicos
- La elección del tipo depende de las necesidades del programa
- Dominar estos tipos mejora la eficiencia y claridad del código

¿Preguntas?

¡Gracias por su atención!

Programación Orientada a Objetos

POO

Programación Orientada a Objetos

Una nueva **forma de pensar**

Programación Orientada a Objetos

Se trata de **descomponer un problema** en subproblemas y más **subproblemas**

Programación Orientada a Objetos

Definir un Dominio del Problema

PROBLEM DOMAIN

Programación Orientada a Objetos

Recopilación de requisitos del cliente y tener por escrito un alcance

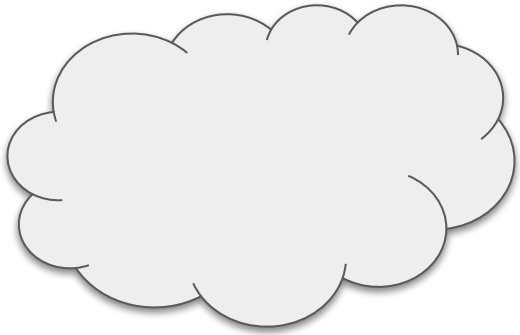
¿Qué queremos lograr?

Programación Orientada a Objetos

Ver un problema como un **escenario** del problema y tratar de **simularlo con objetos**

Identificando Objetos

Pueden ser **Físicos** o **Conceptuales**



Identificando Objetos

Tienen **atributos** (características)

- tamaño
- nombre
- forma
- representan el estado del objeto

Identificando Objetos

Los **nombres** y **atributos** de los **objetos** por lo general son **sustantivos**

`user, session`

Identificando Objetos

Las operaciones suelen ser verbos o sustantivo y verbo

`login, makeReport`

Objeto

Objeto

- Auto

Atributos:

- matricula
- marca
- modelo

Objeto

- Auto

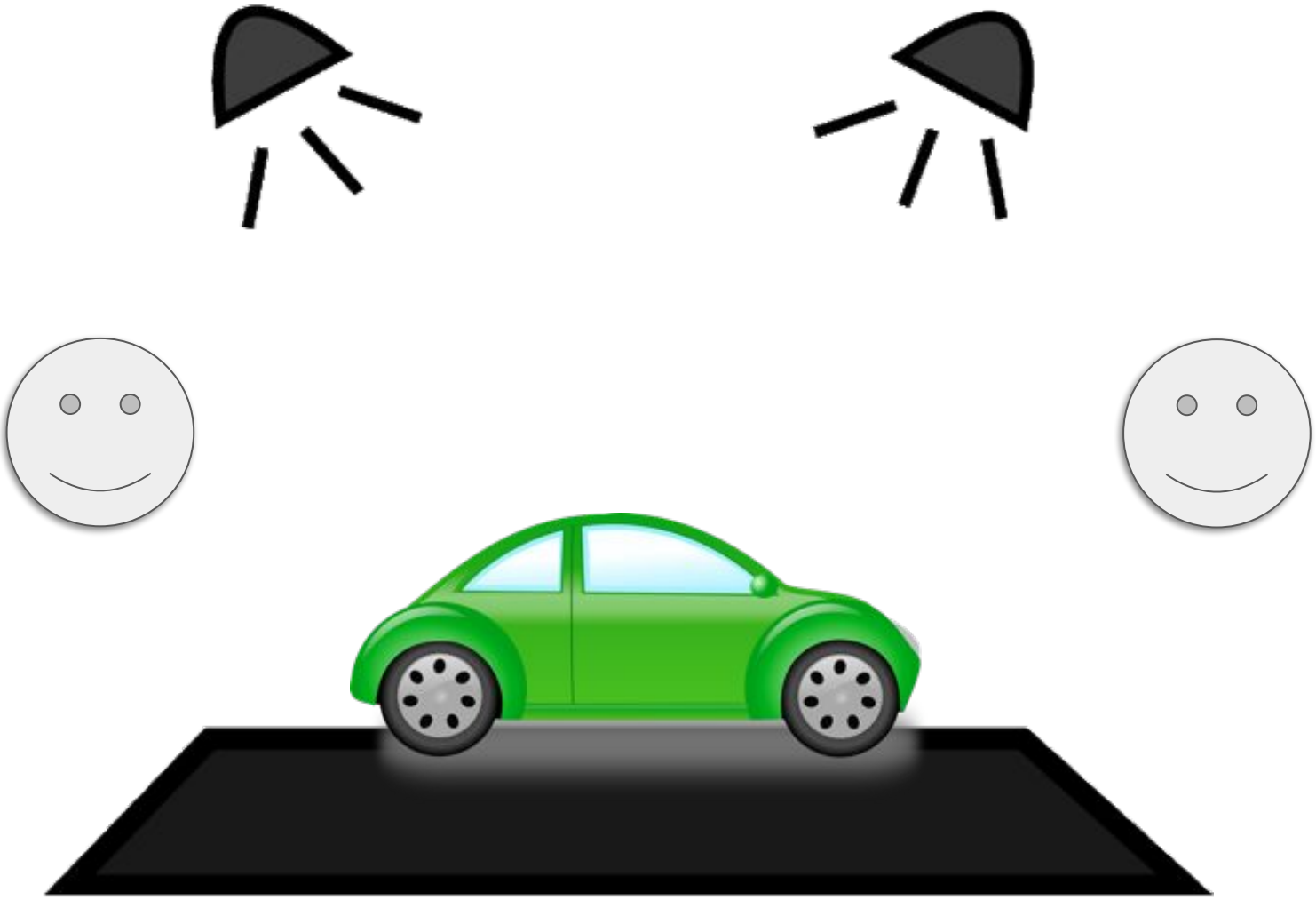
Atributos:

- matricula
- marca
- modelo

Comportamientos:

- arrancar
- frenar
- reversa

Objeto



POO

- Auto

Atributos:

- matricula
- marca
- modelo
- precio
- vendido

Comportamientos:

- mostrarDatos
- esVendido

Clase

Una Clase es la forma en cómo **defines tu objeto** para generar más objetos

Clase

Las Clases son descriptivas son plantillas

Clase

- Auto

atributos
- id
- matricula
- marca
- año
- precio
- vendido

comportamientos
- mostrarDatos
- esVendido

Estructura de una Clase

```
public class Auto {

}
```

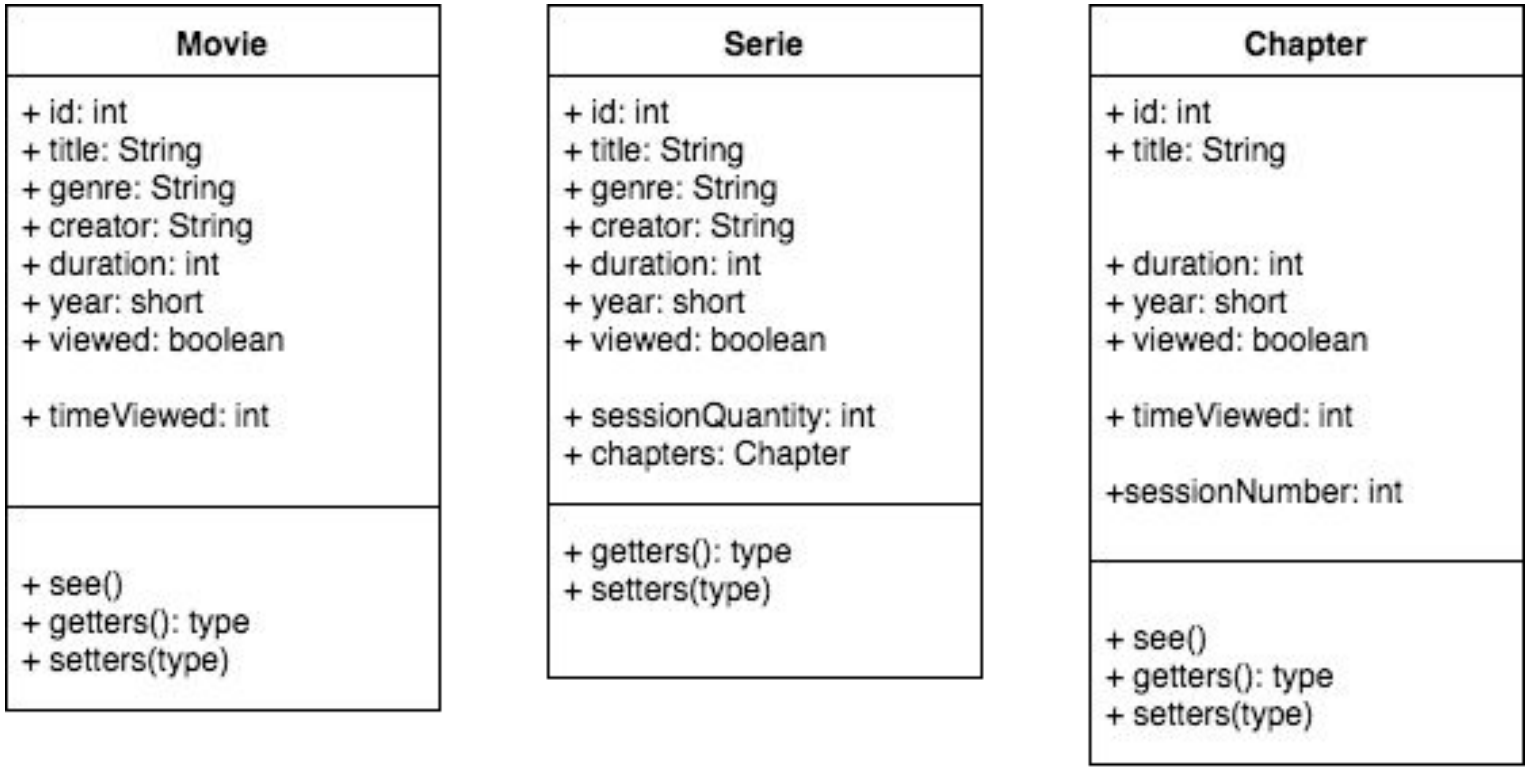
atributos

```
int id;
String matricula
String marca
String modelo
double precio
boolean vendido
```

comportamientos

```
public void mostrarDatos (){
}
public boolean esVendido() {
}
```

Clases

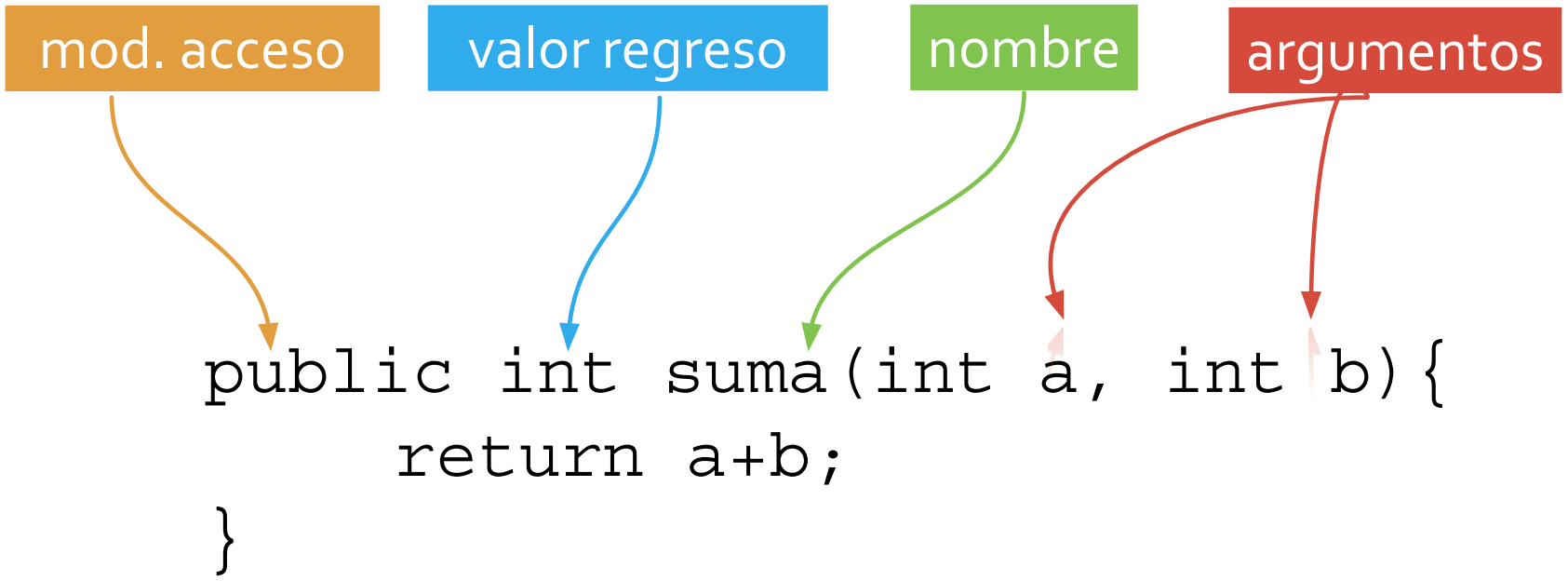

Movie
+ id: int
+ title: String
+ genre: String
+ creator: String
+ duration: int
+ year: short
+ viewed: boolean
+ timeViewed: int
+ see()
+ getters(): type
+ setters(type)
Serie
+ id: int
+ title: String
+ genre: String
+ creator: String
+ duration: int
+ year: short
+ viewed: boolean
+ sessionQuantity: int
+ chapters: Chapter
+ getters(): type
+ setters(type)
Chapter
+ id: int
+ title: String
+ duration: int
+ year: short
+ viewed: boolean
+ timeViewed: int
+sessionNumber: int
+ see()
+ getters(): type
+ setters(type)

Clases

Book
+ id: int + title: String + editionDate: Date + editorial: String + autores: String[] + isbn: String + readed: boolean + timeReaded: int
+ getters + setters + read

Magazine
+ id: int + title: String + editionDate: Date + editorial: String + autores: String[]
+ getters + setters

Métodos


mod. acceso
valor regreso
nombre
argumentos
public int suma(int a, int b){
return a+b;
}

Métodos

Tiene un **valor de regreso** explícitamente invocado en su cuerpo usando la palabra reservada return.

Métodos

No regresa ningún valor si es declarado **void**.

No puede declararse **dentro de otro método**.

Declarando un objeto

```
Auto    miAuto;
```

Tipo de Objeto

Nombre del Objeto

Instanciando un objeto

```
miAuto = new Auto();
```

Nombre del Objeto

Creando el objeto

Declarando e instanciando un objeto

```
Auto miAuto = new Auto();
```

Declarando el objeto

Instanciando/Creando el objeto

Método Constructor

```
miAuto = new Auto();
```

Método Constructor

Método Constructor

Crea nuevas **instancias** de una clase.

Tiene el **mismo nombre** que la clase que inicializa.

Usa la palabra reservada **new** para invocarlo.

Método Constructor

Usa **cero o más argumentos** contenidos dentro de los paréntesis que siguen al nombre.

No regresa un valor.

Utilizando el objeto

```
Auto miAuto = new Auto();
miAuto.marca = “Ferrari”;
miAuto.mostrarDatos();
```

Static

Final

Sobrecarga

Modificadores de Acceso

Static

Accesando a Métodos

```
Auto miAuto = new Auto();
miAuto.mostrarDatos();
```

```
Math.random();
Math.sqrt(25);
Math.PI;
```

Métodos static

Se puede usar en toda la clase

Está definido por la palabra reservada **static**.

Métodos static

Puede ser **accesado indicando el nombre de la clase**, la notación punto y el nombre del método.

Métodos static

Se invoca en una clase que **no tiene instancias de la clase**.

Métodos static

```
public class Calculadora {

    public static int suma(int a, int b){
        return a+b;
    }
}
```

```
Calculadora.suma(5,2);
```

Métodos static

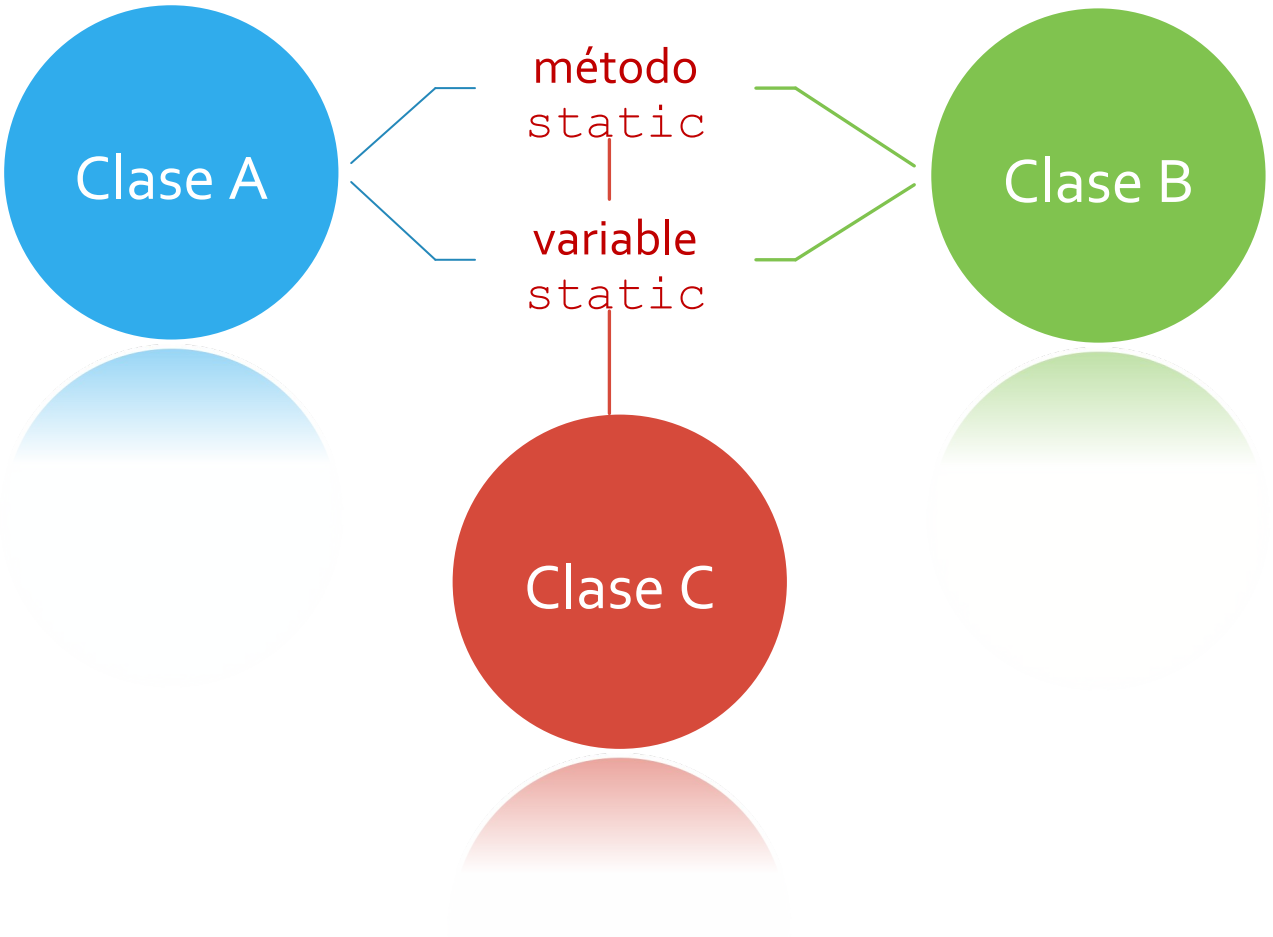
Puede ser invocado en una clase que **no tiene instancias de la clase**.

Miembros static

```
public class Calculadora{

    public static final double PI = 3.1415926

    public static int valor = 0;

}
```

```
Calculadora.PI;

Calculadora.valor;
```

Miembros static



Miembros static

```
import static com.anncode.operaciones.Calculadora.*
import static java.lang.Math.*;

public class Principal{

    public static void main(String[] args){
        System.out.println(suma(3, 5));
        System.out.println(PI);
    }


}
```

Final

Miembros final

```
public class Calculadora{

    public static final double PI = 3.1415926


}
```

```
Calculadora.PI;
```

Sobrecarga

Sobrecarga

A veces necesitamos que dos o más métodos **tengan el mismo nombre pero con diferentes argumentos**

Sobrecarga

```
public class Calculadora{

    public int suma(int a, int b){
        return a+b;
    }

    public float suma(float a, float b){
        return a+b;
    }

    public float suma(int a, float b){
        return a+b;
    }

}
```

Sobrecarga de Constructores

La sobrecarga de constructores se usa para **inicializar objetos**

Sobrecarga de Constructores

```
Auto miAuto = new Auto();
```

Método Constructor

Sobrecarga de Constructores

```
public class Auto {
    int id;
    String matricula
    String marca
    String modelo
    double precio
    boolean vendido
    public Auto(){

    }

    public Auto(int id, String matricula, String precio){
        this.id = id;
        this.matricula = matricula;
        this.precio = precio;
    }
}
```

Modificadores de Acceso

Sobrecarga de Constructores

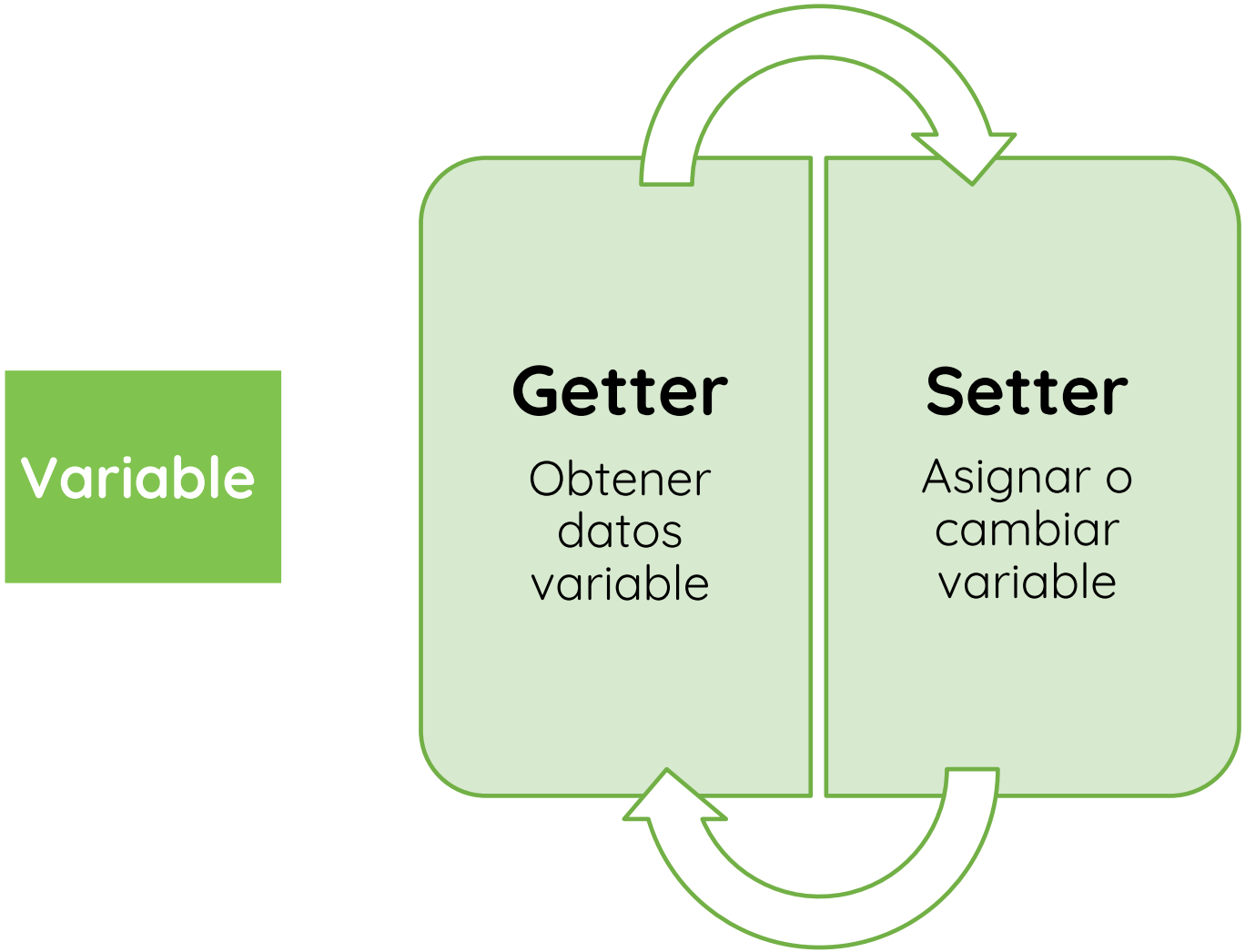
Modificador	Clase	Package	Subclase	Otros
public	✔	✔	✔	✔
protected	✔	✔	✔	●
default	✔	✔	●	●
private	✔	●	●	●

Getters y Setters

Getters y Setters

Leer/Escribir
específicamente los valores
de las **variables** miembro.

Getters y Setters



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnólogo en desarrollo de videojuegos.
- **Código del Programa de Formación:** *217307*.
- **Nombre del Proyecto:** Construcción de software integrador de tecnologías orientadas a servicios
- **Fase del Proyecto:** Analisis.
- **Actividad de Proyecto:** Planificar la interactividad y funcionalidad del nivel.
- **Competencia:** 220501068-Desarrollar la lógica y mecánica del videojuego de acuerdo con el diseño establecido.
- **Resultados de Aprendizaje Alcanzar:**
  - 22050106803 - Programar la interactividad de acuerdo con el diseño del videojuego.
  - 22050107403 - Determinar las características del videojuego de acuerdo con el concepto.
- **Duración de la Guía:** xx horas

2. PRESENTACIÓN

Bienvenido a la actividad del proyecto "Planificar la interactividad y funcionalidad del nivel", donde se proporcionan lineamientos y conocimientos fundamentales sobre el desarrollo de videojuegos. En esta etapa, definiremos el "One Page Document", un documento de una sola página que muestra las características esenciales de la idea inicial de un videojuego de manera práctica y visual.

Se abordarán los fundamentos de la lógica de programación y algoritmos, que son conjuntos de pasos o instrucciones lógicamente ordenadas para resolver problemas o realizar tareas específicas. Esta fase es previa a la programación y esencial antes de escribir las líneas de código que ejecutarán las acciones en el videojuego.

El objetivo de esta guía es transformar ideas en proyectos reales de videojuegos, un proceso emocionante y desafiante. A través de este aprendizaje, desarrollarás habilidades en planificación, organización, creatividad y resolución de problemas. Podrás guiar tu propio proceso y trabajar de manera autónoma y sistemática, promoviendo la construcción significativa de conocimientos y el trabajo colaborativo para el crecimiento integral del grupo. ¡Empecemos! A continuación, se detallan las acciones de aprendizaje necesarias para alcanzar estos objetivos.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es)**

3.1 Actividades de reflexión inicial:

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

RAP-22050107403-AA1-EV1 – Reconocer la Historia y conceptos de diseño del videojuego.

Descripción de la actividad:

Conceptualice, defina los siguientes temas:

- **Historia de los videojuegos:** revise los recursos audiovisuales sugeridos por el instructor y complemente con búsqueda de internet y desarrolle una línea de tiempo con los aspectos mas importantes sobre la historia de los videojuegos.
- **Tipos de videojuegos.** Identifica y define los tipos de juegos que exiten, luego describa las carateristicas que tiene cada tipo desde el punto de vista de los siguientes términos:
  - Historia
  - Personajes
  - Escenarios
  - Niveles
  - Objetos
  - Reglas
  - Mecanicas
- **Sistemas de clasificación en los videojuegos(Audiencia Rating):** realice una búsqueda de información, conceptualice, la clasificación **ESRB y PEGI**.
- **Dispositivos de videojuegos:** Realizar un cuadro comparativo con los 10 dispositivos mas populares del mercado identificando los siguientes atributos:
  - Tipo
  - Generación
  - Desarrollador
  - Fabricante
  - Fecha de lanzamiento
  - Costo
  - Unidades vendidas
  - Datos técnicos
  - Peso
  - Procesador
  - Memoria
  - Almacenamiento
  - GPU
  - Conectividad
  - Vídeo
  - Audio
  - Métodos de entrada
  - Batería
  - Soporte
  - Software
  - Sistema operativo

- Servicio en línea
- Videojuego más vendido.

- **Plataformas de distribución y ejecución actuales de videojuegos:** Identifica y crea un cuadro donde se liste las plataformas de distribución y ejecución de videojuegos.

- **Estilo artístico:** Realice una búsqueda de información, conceptualice, defina los estilo artístico existentes para 2D, 2.5D y 3D.

- **Mecanicas en los videojuegos:** Defina el concepto e indica los elementos que le permitan determinarlas, que tipos existen e identifique juegos que le permitan mostrar ejemplos.

- **Dinamicas en los videojuegos:** Defina el concepto e indica los elementos que le permitan determinarlas, que tipos existen e identifique juegos que le permitan mostrar ejemplos.

- **Pipeline del videojuego:** reconozca de forma conceptual las etapas o fases en el proceso de desarrollo de videoJuego.

- **Niveles en videojuegos**: Define que son los niveles, funciones principales de los niveles, como se clasifican e identifique juegos que le permitan mostrar ejemplos.

- ¿Que son los escenarios?

- ¿Que son los props?

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Taller, Panel, Foro de discusión.

Materiales de formación:

- Historia y conceptos de diseño del videojuego
- Mecánica en los videojuegos
- Dinámicas en los videojuegos
- Estilos artísticos de los videojuegos
- Escenarios y props

Material de apoyo: No Aplica.

Duración de la actividad: 8 horas.

Evidencias de aprendizaje: Documento aplicando normas APA 7th edición

RAP-22050106803-AA2-EV1-identifica los elementos básicos de lógica de programación: fundamentos de programación y diagramas de flujo- Mapa mental sobre Concepto de algoritmos.

Descripción de la actividad:

El aprendiz de forma individual realizará la lectura del material IntConAlg-Diagrama de Flujo.pdf y basado en los concepto ahí consignados creara un mapa mental.

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Taller, Panel, Foro de discusión.

Materiales de formación:

- 1-AP02_IntConAlg-Alg-Dia-Var
- 2-ELEMENTOS DE UN DIAGRAMA.pdf

Material de apoyo: No Aplica.

Duración de la actividad: 4 horas.

Evidencias de aprendizaje: Documento mapa mental sobre conceptos de algoritmos.

3.3 Actividades de apropiación:

RAP-22050107403-AA1-EV2 - Generar el documento inicial de diseño del juego definiendo el one page document acuerdo a la idea interactiva o videojuegos definido.

Descripción de la actividad:

El one page document consiste en un documento de una sola página que muestra las características más importantes de tu videojuego de forma práctica y visual. Esta herramienta es usada en el proceso de diseño para ayudarnos a empezar y comprender nuestro juego, así como para comunicar lo más importante de tu juego a personas cercanas o incluso inversores. La información presentada debe ser muy gráfica, reflejando el estilo artístico del juego, y debe estar ordenada y limpia para una mejor comprensión.

- **Indicaciones**
  - Definir un Equipo de trabajo Maximo 5 personas, puede ser de libre elección o definido por el instructor.
  - El equipo de trabajo redactará el **one page document** el cual consiste que una sola pagina resuman las carateristicas mas importantes del videojuego o experiencia interactiva, este debe ser diseñado como un cavas para compartirlo de forma practica y visual.
  - Los puntos que debe tener este documento son los siguientes:

    1. Nombre del juego:

2. Información del juego
    - Genero:
    - Dispositivos:
    - Audiencia:
    - Rating:
    - Estilo artístico:
    - Fecha de publicación:

3. Resumen: Un pequeño resumen de la historia o el oncepto del juego o experiencia interactiva.
4. Frase de impacto: Un eslogan o frase que represente tu juego o experiencia interactiva.
5. Aspectos Únicos: Aspectos que diferencian tu juego de la competencia.

Nota: Este producto resulta del taller de conceptualización **RAP-22050107403-AA1-EV1 – Reconocer la Historia y conceptos de diseño del videojuego,** realizado anteriormente.

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Proyectos.

Materiales de formación:

Que es una One Page: https://educated-show-144.notion.site/Gu-a-completa-para-crear-un-One-Page-document-efectivo-8cc4498f9d0f46ee92b1bf4daaba74ad?pvs=4

Material de apoyo:

Evidencias de aprendizaje: **one page document[Idea Videojuego].pdf**

Instrumentos de evaluación:

Duración de la actividad: 8 horas.

RAP-22050106803-AA2-EV2-Aplicar el proceso metodologico de análisis y diseño de algoritmos a problemas planteados, según casos asignados.

Descripción de la actividad:

Esta actividad de desarrollo individual tiene como objetivo que el estudiante aplique el proceso metodológico completo de análisis y diseño de algoritmos a problemas específicos asignados. El estudiante deberá abordar cada caso siguiendo una estructura de solución bien definida, que consta de dos partes principales:

1. **Análisis del algoritmo:** a) Reformulación del problema b) Resultados esperados c) Datos disponibles d) Restricciones e) Procesos

2. **Diseño del algoritmo:** a) Entrada b) Proceso c) Salida

El estudiante deberá trabajar de forma individual, aplicando los conocimientos adquiridos en clase para analizar cuidadosamente cada problema asignado, siguiendo paso a paso la estructura de solución proporcionada. Se espera que el estudiante desarrolle habilidades críticas en la descomposición de problemas, identificación de requisitos y diseño de soluciones algorítmicas eficientes.

Esta actividad permitirá al estudiante fortalecer su capacidad de pensamiento lógico y estructurado, así como mejorar sus habilidades en la creación de algoritmos bien diseñados y documentados. Al finalizar, el estudiante habrá ganado experiencia práctica en la aplicación sistemática de métodos de análisis y diseño de algoritmos a problemas del mundo real.

Indicaciones:

1. En clase el aprendiz debe plantear un problema de la vida cotidiana al cual debe aplicar los pasos de análisis y diseño de algoritmos explicados en el material didáctico suministrado por el instructor.
2. Realizar los ejercicios definidos por el instructor en el taller problemas de algoritos: https://educated-show-144.notion.site/Ejercicios-de-Algoritmos-2cf55d2d696b4f86ba289a9042482075?pvs=4 definido, esta solución se debe presentar en un documento de Word o Google docs, aplicando las normas apa 7th y la estructura del documento minimamente debe tener los siguientes apartados:
    - *Presentación (Portada)*
    - *Introducción*
    - *Desarrollo del trabajo*
    - *Conclusiones*

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Aprendizaje Basado en problemas.

Materiales de formación:

- 1-AP02_IntConAlg-Alg-Dia-Var

Material de apoyo:

- Explicación-Proceso de análisis y desarrollo de algoritmos: https://educated-show-144.notion.site/Explicaci-n-Proceso-de-an-lisis-y-desarrollo-de-algoritmos-b71ae5e933194867bcbad04b7d1f2667
- Ejemplo-Proceso de análisis y desarrollo de algoritmos: https://educated-show-144.notion.site/Ejemplo-Proceso-de-an-lisis-

y-desarrollo-de-algoritmos-e1b03a664bfc411aa63adefbad654ef5?pvs=4

Evidencias de aprendizaje: Documento aplicando normas APA 7th edición análisis y diseño de algoritmos.doc

Instrumentos de evaluación: Lista de chequeo

Duración de la actividad: 12 horas.

3.4 Actividades de Transferencia el Conocimiento:

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación

5. GLOSARIO DE TÉRMINOS

- **Algoritmo**: conjunto de pasos que se llevan a cabo para dar solución a un problema.

6. REFERENTES BILBIOGRÁFICOS

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Carlos Enrique Navia Torres**	**Instructor**	**CCyT**	**Julio de 2024**

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnólogo en Análisis y Desarrollo de Software.
- **Código del Programa de Formación:** *228118*.
- **Nombre del Proyecto:** Construcción de software integrador de tecnologías orientadas a servicios
- **Fase del Proyecto:** Planeacion.
- **Actividad de Proyecto:** .
- **Competencia:** Determinar los requisitos funcionales y no funcionales del software de acuerdo con los requerimientos del cliente
- **Resultados de Aprendizaje Alcanzar:**
    - Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico.
- **Duración de la Guía:** 36 horas

2. PRESENTACIÓN

El SENA le extiende una cordial bienvenida a la guía de aprendizaje que comprende el resultado de aprendizaje "Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico". En este contexto, la recolección de requerimientos cumple un papel primordial en el proceso de desarrollo de software: la definición de lo que se desea producir.

Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. De esta manera, se pretende minimizar los problemas relacionados con el desarrollo de sistemas.

Por consiguiente, se presentan a continuación cada una de las acciones de aprendizaje que le permitirán desarrollar lo anteriormente mencionado.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es)**

3.1 Actividades de reflexión inicial:

Descripción de la actividad:

El diseño y desarrollo de un sistema de información normalmente está orientado a brindar una solución particular ya sea compleja o sencilla, de un problema o una necesidad, en los diversos sectores productivos los sistemas de información se han convertido en una herramienta fundamental para la ejecución y control de procesos operativos, administrativos y gerenciales acordes con los objetivos empresariales de manejo, organización y flujo de información; por lo anterior, y antes de dar inicio a las actividades de aprendizaje, es importante que el aprendiz en formación, identifique los elementos que intervienen en el diseño y desarrollo de soluciones informáticas, y los saberes previos que sobre la temática posee o descubra, para ello se hace necesario, y como parte de la fundamentación requerida, que el aprendiz realice una reflexión en torno a los siguientes cuestionamientos:

- ¿Qué utilidad presta la clara definición de un problema o necesidad, y el planteamiento de su objeto de estudio para definir la solución a desarrollar?
- Para realizar una recopilación de información y clasificarla adecuadamente con miras a la definición de requerimientos de un sistema de información: ¿Qué procedimientos cree que son necesarios para su recolección, y con qué instrumentos?
- ¿Qué dificultades u obstáculos considera pueden darse en la identificación de requerimientos para poder definir las funcionalidades de un sistema de información?
- ¿Qué entiende por requerimientos funcionales de un sistema de información?
- ¿Sabe usted para qué sirve el determinar las funcionalidades de un sistema de información, al momento de intentar dar solución a una necesidad o problema en particular?
- ¿Conoce alguna técnica o herramienta para determinar las funcionales de un sistema de información?
- El identificar las posibles funcionalidades de una solución informática, ¿qué ventajas aporta en las actividades de análisis, diseño, desarrollo e implementación de un sistema de información?

Luego de que cada aprendiz de forma individual responda los cuestionamientos planteados anteriormente en los equipos de trabajos establecidos, unificar los conceptos productos de las respuestas resueltas individualmente.

Finalmente el equipo designa a dos represéntates que tendrá el rol de panelista para la actividad final panel conversatorio.

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Panel, Foro de discusión.

Materiales de formación:

Material de apoyo:

Duración de la actividad: 8 horas.

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

Descripción de la actividad:

Para lograr el desarrollo de las actividades de aprendizaje que se propone, es necesario que el aprendiz realice la lectura de los materiales de formación y se apropie de los contenidos y conocimientos dispuestos en los diversos recursos que se encuentran en plataforma virtual, en cada objeto de aprendizaje (OA), encontrará una actividad didáctica como instrumento de realimentación que le permitirá verificar el alcance de conocimientos adquiridos, podrá revisar el mapa orientador de contenidos, analizar el glosario de términos relevantes al tema de estudio, cotejar la bibliografía para la exploración y seguimiento de la temática desarrollada y estudiar los contenidos en general.

1. OA. Técnicas de recolección de información.
2. OA. Mapa de procesos.
3. OA. Gestión por procesos y calidad.
4. OA. Ingeniería de Requerimientos.
5. OA. El proceso de software.

En equipo de trabajo se asignata un tema en el que el grupo realizará un presentación en formato de una microclase donde los integrantes presentaran de forma teórica y práctica los conceptos y procedimientos de cada tema asignado en el ambiente de formación a sus compañeros, al final los presentadores realizaran un ejercicio práctico de los temas a sus compañeros, que consiste en el desarrollo de ejercicios los cuales deben ser desarrollados por sus otros compañeros aprendices

Evidencia (De Desempeño)

GA1-220501092-AA1-EV2 - Presentación en PowerPoint con la estructuración de los temas y Taller de ejercicios prácticos del tema.

Evidencia (De Conocimiento)

GA1-220501092-AA1-EV3 - Cuestionario Técnico AP1. Con el fin de obtener evidencia de esta actividad de apropiación, con relación a los conocimientos adquiridos por el aprendiz para el logro de los resultados de aprendizaje previstos, mediante la interacción de éste con el material de estudio

suministrado en cada uno de los objetos de aprendizaje dispuestos en plataforma, se dispone de un cuestionario que determina si el aprendiz maneja adecuadamente los conceptos sobre técnicas de recolección de información, mapa de procesos, redes, sistemas operativos, el proceso de software, gestión por procesos y calidad.

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas:

Materiales de formación:

1. OA. Técnicas de recolección de información.
2. OA. Mapa de procesos.
3. OA. Gestión por procesos y calidad.
4. OA. Ingeniería de Requerimientos.
5. OA. El proceso de software.

Material de apoyo:

Duración de la actividad: 8 horas.

3.3 Actividades de apropiación:

GA1-220501092-AA2- EV01-Identificación de la necesidad de sistemas de información.

Descripción de la actividad:

Con el fin de obtener la necesidad o problema a solucionar como proyecto de sistemas de información, se debe elaborar y aplicar dos instrumentos (entrevista y encuestas) con base a las técnicas de recolección de datos apropiadas, para generar la identificación de la necesidad o problema solucionar en el curso. Para ello se debe identificar en una empresa una necesidad y con ayuda del instructor aplicar correctamente las técnicas de recolección de datos, y sus resultados se deben analizar obteniendo así la situación problema y la idea de solución a través de un sistema de información.

DESCRIPCIÓN DE LA EVIDENCIA

Con base en las técnicas aprendidas en el Objeto de Aprendizaje “Técnicas de recolección de Datos”, disponible en el material de estudio, el equipo de trabajo del proyecto genera un documento en el que registre lo siguiente:

- Ámbito empresarial
- Diseño de instrumento entrevista
- Entrevistas realizadas
- Identificación de la necesidad o situación problema

- Idea de solución
- Diseño de instrumento encuesta
- Validación de la idea de solución
- Ajustes de la idea

1. **Identificar el ámbito empresarial**: en este paso se debe identificar una empresa, puede ser en la que labore el aprendiz o la empresa de algún amigo o familiar, como requisito es que debe estar legalmente constituida y clasifican todas las empresas desde micro hasta grandes empresas.

   En esta paso el equipo de trabajo debe identificar una empresa la legalmente constituida para iniciar con el proceso Diagnostico y Recolección y análisis de Información.
   Para la idenficacion se sugiere al equipo hacer una lluvia de ideas de las posibilidades que se tienen.
2. **Diseñar Instrumentos:** en este paso el equipo diseñará el instrumento de la entrevista a realizar de donde deberá obtener la información completa de la empresa, número de empleados, misión, visión entre otros datos de la empresa y obtener la situación problema que se puede solucionar con un sistema de información. ejemplo: "tenemos un problema de inventarios que no hemos podido solucionar por qué…". Para ello debe diseñar la entrevista y las preguntas a realizar, seleccionar a 3 personas de la empresa donde crea que exista una situación problema. También deberá con el instructor compartir el diseño antes de la ejecución de la entrevista con el fin de aclarar las preguntas a realizar y de cómo realizar la entrevista de la mejor manera teniendo en cuenta lo aprendido en las técnicas de recolección de datos.
3. **Ejecutar las entrevistas:** seleccionadas las personas y ya con el instrumento elaborado el aprendiz realizará las entrevistas, consignando en un video cada una con el fin de obtener la evidencia y repasar sobre las respuestas de los entrevistados.
4. **Describir la situación problema e identificarlo**: en este paso el aprendiz aplicará el análisis de las entrevistas y determinará una situación problema que será la que se solucionará a través del sistema de información. Una vez descrita la situación problema el aprendiz compartirá con su instructor y este le apoyará para depurar aún más la problemática y la validará en caso de ser apropiada. De lo contrario sugerirá al aprendiz ampliar más la situación.
5. **Plantear una solución desde el punto de vista de sistemas de información:** el aprendiz planteará a manera de *idea*, como un sistema de información puede solucionar el problema identificado.
6. **Validar la solución o idea:** con el problema obtenido del análisis anterior y la idea de solución se debe elaborar un cuestionario para validar si esa idea es necesaria y se ajusta a la solución del problema. Diseñara este instrumento con la asesoría del instructor antes de la aplicación del cuestionario a través de la encuesta definiendo lo que se quiere medir. Un ejemplo de pregunta en la encuesta sería: "*Usted cree que la idea propuesta puede solucionar el problema…"*. Esta se puede realizar a través de herramientas TIC o en forma física a una muestra de la población. si la empresa es pequeña se aplicará a toda la empresa. Si la empresa es mediana o grande a una población aplicando lo aprendido en la técnica de encuestas y muestreo.

7. **Tabular la información:** con las encuestas realizadas deberá cuantificar los resultados a través del proceso de tabulación y generar un informe de validación de la idea donde se podrá analizar qué se debe mejorar de la idea o si por el contrario la idea propuesta está alineada con la situación problema.

Ambiente requerido: Equipos de computo, conexión a Internet

Estrategias o técnicas didácticas activas: Aprendizaje Basado en problemas

Materiales de formación: OA. Técnicas de recolección de información.

Material de apoyo: Template Entrevistas.docx

Evidencias de aprendizaje:

GA1-220501092-AA2- EV01-Identificación de la necesidad de sistemas de información. Para cumplir con esta evidencia, es importante que haya realizado la actividad de apropiación referida a la comprensión al material de estudio presentando en esta guía. De acuerdo con las indicaciones de su instructor.

Los documentos asociados a la evidencia son los siguientes: siga las pautas y orientaciones consignadas en los siguientes documentos para el adecuado desarrollo de la evidencia.

1. **Documento realizado en procesador de texto, con los resultados de los instrumentos realizados.**

   El documento deberá contener un informe con el resultado del paso a paso ejecutado con los instrumentos de recolección de información realizados y la necesidad identificada.

2. **Videos o grabación de las entrevistas.**

   Videos o grabación de las entrevistas realizadas en formato mp3 o mpg4

Instrumentos de evaluación:

Instrumento de evaluación: IE-AP01-AA1-EV01-Identificación-Necesidad-SI.

Duración de la actividad: 16 horas.

3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

Ambiente requerido:

Estrategias o técnicas didácticas activas:

Materiales de formación

Material de apoyo:

Evidencias de aprendizaje:

Instrumentos de evaluación:

Duración de la actividad: horas.

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación

5. GLOSARIO DE TÉRMINOS

- **Encuesta**: Conjunto de preguntas tipificadas dirigidas a una muestra representativa, para averiguar estados de opinión o diversas cuestiones de hecho.
- **Entrevista**: Conferencia entre dos o más personas en lugar determinado, sustentada en una lista de preguntas estructuradas con el objetivo de obtener información.
- **Entrevistado**: Persona a quien va dirigida la entrevista, es quien responde el cuestionario.
- **Entrevistador**: Persona encargada de preparar, aplicar y analizar la entrevista.
- **Estructura**: Distribución adecuada de un grupo de preguntas para ser aplicadas en una entrevista.
- **Categorización**: consiste en la recolección y clasificación de la información de manera ordenada.
- **Investigación**: Su fin es ampliar el conocimiento científico, sin perseguir, en principio, ninguna aplicación práctica.
- **Método**: Modo en que se procede y los procedimientos seguidos para hallar la verdad acerca del objeto de investigación.
- **Metodología**: ciencia que consta de métodos y técnicas, que se aplican sistemáticamente durante un proceso de investigación o para solucionar una problemática.

- **Población**: Conjunto de elementos o personas sobre los que se está interesado en realizar una observación para obtener información relevante para el sistema de información.
-

6. REFERENTES BILBIOGRÁFICOS

- Gómez, M. (2006). *Introducción a la metodología de la investigación científica*. Córdoba, Argentina: Brujas.
- Kendall, K. &. (2011). *Análisis y Diseño de Sistemas de información*. México: Pearson.
- Pressman, R. (2010). *Ingeniería del software, un enfoque práctico* (Séptima edición). Bogotá: McGraw-Hill.
- P. Bourque y R.E. Fairley, (eds) (2014)., *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society.
- Rodríguez Mogel, E. (2005). *Metodología de la investigación*. Villahermosa - México: Universidad de Juares, Autónoma de Tabasco.

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Carlos Enrique Navia Torres**	**Instructor**	**CCyT**	**Julio de 2024**

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

Brief para Coffee Tasting VR Experience.

1. Objetivo

Crear una experiencia de realidad virtual inmersiva y educativa que simule el proceso de catación de café profesional, permitiendo a los usuarios aprender y practicar las técnicas de evaluación sensorial del café en un entorno virtual preciso y detallado.

2. Premisa (Propósito)

1. "Coffee Tasting VR" es una aplicación que transporta a los usuarios a un laboratorio de catación de café virtual, donde pueden perfeccionar sus habilidades de evaluación sensorial sin necesidad de equipos físicos costosos.
2. Los jugadores experimentarán el proceso completo de catación de café, desde la molienda de los granos hasta la evaluación final, interactuando con equipos virtuales precisos y muestras de café simuladas.
3. Los temas principales son el aprendizaje experiencial, el desarrollo de habilidades sensoriales y la apreciación del café de especialidad.

3. Público Objetivo

- Baristas en formación
- Profesionales de la industria del café
- Entusiastas del café que desean mejorar sus habilidades de catación
- Estudiantes de gastronomía y ciencias de los alimentos

4. Necesidades clave

1. Proporcionar una plataforma accesible y económica para practicar la catación de café.
2. Ofrecer una experiencia educativa interactiva que simule con precisión el proceso de catación profesional.
3. Permitir a los usuarios mejorar sus habilidades sensoriales y su conocimiento sobre la evaluación del café.

5. Puntos de dolor y limitaciones clave

1. La falta de acceso a equipos profesionales de catación y varietales de café para muchos entusiastas y estudiantes.

2. El costo prohibitivo de participar en sesiones de catación profesional reales.
3. Limitaciones técnicas en la simulación de olores y sabores en VR.

6. Tecnología utilizada

La experiencia está creada para Oculus Quest 2, aprovechando su capacidad de seguimiento de manos sin controladores para una interacción más natural con los objetos virtuales y su capacidad de procesamiento para simular físicas realistas de líquidos y partículas.

7. Narrativa / Trama

Como desarrolladores del proyecto el equipo debe establecer la narrativa siguiendo las siguientes indicaciones y estructura:

> Escribe la narrativa de tu experiencia. Usa un mapa de historia como guía para el arco narrativo y crea un storyboard de la experiencia. *Ver enlaces para recursos y ejemplos abajo. Piensa en cómo los cineastas hacen el storyboard de una película.*
>
> **Escena de establecimiento**
> 1. Acción
> 2. Sonido
> 3. UI
>
> **Paso del viaje 1 … N**
> 1. Acción
> 2. Sonido
> 3. UI
>
> **Conclusión**
> 1. Action
> 2. Sound
> 3. UI

8. Viaje del Usuario (alto nivel)

1. El jugador comienza en el laboratorio de catación virtual.
2. Tutorial guiado sobre el uso de los equipos y el proceso de catación.
3. Preparación de las muestras de café (molienda, pesaje).
4. Evaluación de la fragancia del café molido.
5. Adición de agua y cronometrado del proceso.
6. "Romper la capa" y evaluar los aromas.
7. Degustación y evaluación del café en diferentes temperaturas.

8. Puntuación final y retroalimentación sobre la catación.

9. Diseño del Entorno

Un laboratorio de catación virtual con una mesa central, equipos de catación dispuestos alrededor (balanza, molino, teteras, tazas, cucharas, cronómetro), y estaciones de trabajo individuales. El espacio debe ser luminoso y limpio, similar a un laboratorio profesional.

10. Boceto y Estilo de UI

UI minimalista e intuitiva con indicadores holográficos para temperatura, peso, tiempo y puntajes. La interfaz debe ser no intrusiva, permitiendo una visión clara del entorno y los objetos de catación. Se utilizarán paneles flotantes para información detallada y formularios de evaluación.

11. Mecánicas del Juego

Mecánica 1: Interacción Precisa con Objetos (MDA)

1. Nombre de la Mecánica

Manipulación de Precisión

2. Descripción Breve

Permite a los usuarios interactuar con objetos virtuales con alta precisión, simulando las acciones reales en una sesión de catación de café.

3. Metodología Utilizada

[X] MDA [ ] Diseño Basado en Acciones

4. Desglose de la Mecánica

- Mecánica: Sistema de física detallado para la manipulación de objetos, incluyendo colisiones realistas, gravedad y simulación de líquidos.
- Dinámica: Los jugadores deben realizar movimientos precisos para pesar, moler, verter y manipular el café y los equipos.
- Estética: Sensación de control, precisión y profesionalismo.

5. Controles/Inputs

Seguimiento de manos del Oculus Quest 2 para agarrar, soltar y manipular objetos. Gestos específicos para acciones como "romper la capa" o sorber el café.

6. Progresión/Variaciones

La precisión requerida aumenta con niveles más avanzados, introduciendo muestras más delicadas o equipos más sofisticados.

7. Interacción con Otras Mecánicas

Se integra con el sistema de evaluación sensorial y el cronometrado de procesos.

8. Balanceo

Ajuste de la sensibilidad de las interacciones para equilibrar realismo y facilidad de uso. Implementación de un sistema de ayuda sutil para guiar movimientos precisos.

9. Ejemplos en Juegos Existentes

"Job Simulator" para interacciones básicas, "Eleven Table Tennis" para precisión en movimientos.

10. Prototipo/Prueba

Desarrollar un prototipo enfocado en la manipulación de la balanza y el vertido de agua en las tazas.

Mecánica 2: Evaluación Sensorial (MDA)

1. Nombre de la Mecánica

Análisis Sensorial Virtual

2. Descripción Breve

Simula la experiencia de evaluar las características sensoriales del café, incluyendo aroma, sabor, cuerpo y post-gusto.

3. Metodología Utilizada

[X] MDA [ ] Diseño Basado en Acciones

4. Desglose de la Mecánica

- Mecánica: Sistema de simulación de propiedades sensoriales del café, con variables como intensidad, complejidad y notas específicas.
- Dinámica: Los jugadores deben identificar y evaluar diferentes aspectos del café, tomando decisiones sobre sus características.
- Estética: Descubrimiento, concentración y apreciación.

5. Controles/Inputs

Gestos de olfateo para evaluar aromas, movimientos de sorber para degustar. Interfaz de evaluación para registrar puntuaciones y notas.

6. Progresión/Variaciones

Introducción gradual de perfiles de café más complejos y sutiles. Desafíos de identificación ciega de varietales.

7. Interacción con Otras Mecánicas

Se integra con la manipulación de objetos y el sistema de puntuación.

8. Balanceo

Ajuste de la dificultad de identificación basado en el nivel de experiencia del usuario. Implementación de un sistema de pistas para usuarios novatos.

9. Ejemplos en Juegos Existentes

"Wine Tasting Simulator" para evaluación de características sensoriales, "Parfum" para identificación de aromas.

10. Prototipo/Prueba

Crear un prototipo centrado en la evaluación de aromas, utilizando partículas visuales para representar diferentes notas olfativas.

Mecánica 3: Cronometrado de Procesos (Diseño Basado en Acciones)

1. Nombre de la Mecánica

Gestión Temporal de Catación

2. Descripción Breve

Simula la importancia del tiempo en el proceso de catación, requiriendo que los usuarios realicen acciones específicas en momentos precisos.

3. Metodología Utilizada

[ ] MDA [X] Diseño Basado en Acciones

4. Desglose de la Mecánica

- Acción Principal: Iniciar, detener y monitorear el cronómetro durante diferentes etapas de la catación.
- Sistemas de Soporte: Indicadores visuales y auditivos del paso del tiempo, cambios en las propiedades del café basados en el tiempo transcurrido.
- Retroalimentación: Alertas visuales y sonoras para momentos clave, cambios en la apariencia y propiedades del café.

5. Controles/Inputs

Gestos de la mano para iniciar/detener el cronómetro, interfaz de voz para comandos de tiempo.

6. Progresión/Variaciones

Introducción de múltiples muestras que requieren gestión de tiempo simultánea. Desafíos de velocidad en cataciones avanzadas.

7. Interacción con Otras Mecánicas

Afecta directamente a la evaluación sensorial y la manipulación de objetos, determinando los momentos óptimos para cada acción.

8. Balanceo

Ajuste de los márgenes de error permitidos en el tiempo para diferentes niveles de dificultad. Implementación de un sistema de pausa para usuarios que necesiten más tiempo.

9. Ejemplos en Juegos Existentes

"Overcooked" para gestión de tiempo en múltiples tareas, "Keep Talking and Nobody Explodes" para coordinación basada en tiempo.

10. Prototipo/Prueba

Desarrollar un prototipo que se centre en el proceso de "romper la capa" y evaluar aromas en intervalos de tiempo específicos.

12. Recursos necesarios

Entorno

- Modelo 3D detallado del laboratorio de catación

Accesorios

- Modelos 3D de equipos de catación (balanza, molino, teteras, tazas, cucharas, cronómetro)

Personaje

- Manos del jugador con alta fidelidad y física realista

Sonido

- Efectos de sonido realistas para cada equipo y acción
- Música ambiental relajante

Efectos Visuales

- Simulación de vapor y líquidos
- Partículas para representar aromas

UI

- Interfaces holográficas para puntuación y feedback
- Formulario de evaluación virtual

Animaciones

- Animaciones fluidas para las interacciones con objetos
- Simulación de movimiento de líquidos

13. Inspiración / Referencias

- Estética visual inspirada en laboratorios modernos de catación de café como los de la Specialty Coffee Association.
- Inspiración en documentales sobre catación de café para la atmósfera y el ambiente.
- Juegos de simulación como "Cooking Simulator VR" para la física de objetos y líquidos.

Producto Esperado

Los aprendices deberán crear una aplicación en VR enmarcada en la temática del reto.
Dicha aplicación deberá permitir:

1. Visualizar un ambiente inmersivo (según el tema del reto)
2. Uso de objetos de plugins
3. Modelado de objetos propios
4. Animaciones de dichos objetos
5. Interacciones y respuesta del entorno
6. Interactuar mediante manos virtuales manejadas con los controles del dispositivo Meta Quest 2
7. Permitir interacciones: Agarrar y hacer “teleport”.
8. Activar cambios en el ambiente debidos a eventos producidos por interacciones del usuario (mínimo 3, eg: cambiar un texto, cambiar la apariencia de un objeto o activar una animación, etc).
9. Mostrar información en GUI (e.g: textos en el ambiente, menú inicial o menús de opciones en el ambiente)
10. Ejecutar la aplicación desde un .apk

Entregables

- Video gameplay (máximo 5 mins). En este video se espera poder visualizar qué lograron hacer, frente al listado de funcionalidades esperadas descrito arriba.
- URL para descargar el ejecutable
- Presentación - pitch de 5 minutos

Ítems de evaluación.

1. Visualización del ambiente inmersivo
2. Uso de objetos de plugin
3. Modelado de objetos propios
4. Animaciones de objetos propios
5. Interacciones y respuesta del entorno
6. Mostrar información en GUI
7. Ejecutar la aplicación desde un .apk
8. Pitch
9. Video

CONTROL DEL DOCUMENTO

<SENA /Soft>

Brief plan de entrenamiento **SENA-XR: Desarrollo de apps de realidad extendida**

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Carlos Enrique Navia Torres**	**Instructor**	**Centro de Comercio y Turismo - Regional Quindío.**	**22/01/2024**

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

Información adicional.

Equipos Necesarios para Catar Café

1. **Balanza**
   - Preferiblemente con precisión de 0,1g
   - Para medir 8,25g de café por 150mL de agua (proporción 1:18)
2. **Molino**
   - Dedicado exclusivamente para cataciones, con ajuste fijo
   - Debe moler el café de manera que 70-65% pase por un tamiz #20
3. **Agua**
   - Recién hervida y filtrada
   - Temperatura: 92,2º – 94,4º C
   - Parámetros ideales: 150 ppm TDS, 68 ppm calcio, 40 ppm alcalinidad, 7.0 pH
4. **Tetera**
   - Para servir el agua
5. **Tazas**
   - 5 tazas por muestra (en cataciones simples, 2-3 pueden ser suficientes)
   - Capacidad: 207-266 mL
   - Diámetro superior: 76-89 mm
   - Con tapas
   - Tazas adicionales para enjuagar
6. **Cucharas**
   - Anchas y poco profundas, similares a cucharas de sopa
7. **Cronómetro**
   - Al menos uno
8. **Toallas**
   - Para limpiar derrames y secar cucharas
9. **Mesa de catación**
   - 1 metro cuadrado por cada 6 personas
   - 90 cm de espacio alrededor
   - 1,5 m de radio entre mesas si hay varias
10. **Otros**
    - Lápices o lapiceros
    - Formularios de catación
    - Escupidera o receptáculo para escupir

Nota: Estos requisitos están basados en los protocolos de la SCA (Specialty Coffee Association).

Realización/Ejecución

1. Muela el café inmediatamente antes de catar; si desea moler de antemano, cubra las tazas para preservar las propiedades aromáticas. Antes de moler la muestra de café, muela algunos granos para quitar residuos de otros cafés en el molino.
2. Evalúe la fragancia del café molido.
3. Vierta agua en las tazas (92.2º – 94.4º C o 200º F (+/- 2º F)) y active el cronómetro
4. Espere 4 minutos antes de “romper la capa superior” de la taza
    1. Usando una cuchara, empuje contra la capa de café que se ha formado en la superficie de la taza.
    2. Enjuague la cuchara entre cada taza.
    3. “romper la capa” libera aromas atrapados debajo de la capa de café molido.
    4. Asegúrese de que todos los granos de café estén dispersos y que solo quede una ligera espuma de color claro en la superficie.
5. Remueva la espuma de la superficie de la taza con una o dos cucharas. Tenga cuidado en no agitar el café molido del fondo de la taza.
6. Deje que el café se enfríe. Es común esperar hasta llegar a una temperatura de al menos 75o C, generalmente se requiere 10 o 15 minutos para llegar a esta temperatura. Utilizar el cronómetro ayuda al catador a decidir cuando está bien empezar la evaluación del licor. Probar el café varias veces mientras que se sigue enfriando, permite ver los cambios asociados con la temperatura y la extracción, y esto puede proveer más información sobre el carácter del cafe.
7. Sorber el café al degustar se ha convertido en parte de la norma. Incorporar aire al respirar tiene varios beneficios:
    1. Enfría el café un poco antes de que toque el paladar.

2. Puede que sea más fácil identificar sabores complejos cuando se incorpora oxígeno.
3. Permite que el café cubra el paladar completo de una vez.

8. Se sugiere escupir el licor , ya que el exceso de cafeína puede ser desagradable y afectar la capacidad de evaluar las muestras.
9. Es beneficioso anotar sus impresiones y puntajes durante la catación, comúnmente se usa un formulario de evaluación.
    1. El café se evalúa en una escala de 100 puntos con los siguientes rangos:
        1. < 69,9 = defectuoso
        2. 70 79,9 = calidad comercial
        3. 80+ = café de especialidad
        4. 90+ = café superior
    2. Las discusiones grupales pueden ayudar a calibrar al grupo de catadores. Es aceptable volver a una taza para probar el café nuevamente, esta vez con los comentarios de otros catadores en mente.