LENGUAJES DE PROGRAMACION

LAURA VALENTINA CANO PADILLA

ANA MARIA LOZANO

1002

I.E.D SAN JOSEMARIA ESCRIVA DE BALAGUER

CHIA: CUNDINAMARCA

INFORMATICA

2014

LENGUAJES DE PROGRAMACION

LAURA VALENTINA CANO PADILLA

ANA MARIA LOZANO

1002

FRANCISCO PINZON

I.E.D SAN JOSEMARIA ESCRIVA DE BALAGUER

CHIA: CUNDINAMARCA

INFORMATICA

2014

Taller # UNO (1) TERCER PERIODO

Introducción:

Por medio de este taller, querido visitante sé le da a conocer los lenguajes de programación, conceptos básicos, clasificación, ventajas y desventajas.

Objetivos:

-Averiguar y conocer nuevos términos de programación.

-Dar a conocer al visitante lenguajes de programación, conceptos básicos y sus clasificaciones.

-Aprender lenguajes de programación.

-Dar a conocer una herramienta de aprendizaje acerca del área de informática

Índice:

1. Definir los siguientes términos:

-Proceso

-Actividad

-Programar

-Programación de computadores

-Lenguaje binario

-Lengua assembler

2. ¿Cómo se clasifican los lenguajes de programación?

3. ¿Que son los lenguajes de alto, medio y bajo nivel?, dar tres ejemplos de cada uno.

4. Hacer un cuadro comparativo que permita observar ventajas y desventajas de los lenguajes de alto, medió y bajo nivel

5. ¿Que son las palabras reservadas en programación?

6. ¿Qué es la sintaxis de un lenguaje de programación?

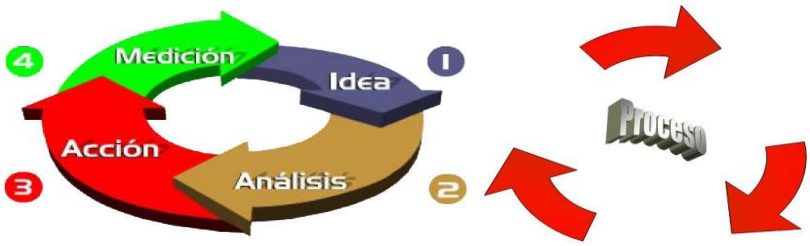
7. ¿Que son los lenguajes de quinta generación?

8. Buscar 5 programas que permita la creación de juegos y hacer cuadro comparativo de ellos con ventajas y desventajas.

Desarrollo

1. Punto –Definición términos

Proceso: Se denomina proceso a la consecución de determinados actos, acciones, sucesos o hechos que deben necesariamente sucederse para completar un fin específico. Todos estos pasos o instancias que componen un proceso deben ser organizados, coordinados y realizados de manera sistemática.



Actividades: Es el conjunto de acciones que se llevan a cabo para cumplir las metas de un programa o subprograma de operación, que consiste en la ejecución de ciertos procesos o tareas, (mediante la utilización de los recursos humanos, materiales, técnicos, y financieros asignados a la actividad con un costo determinado).

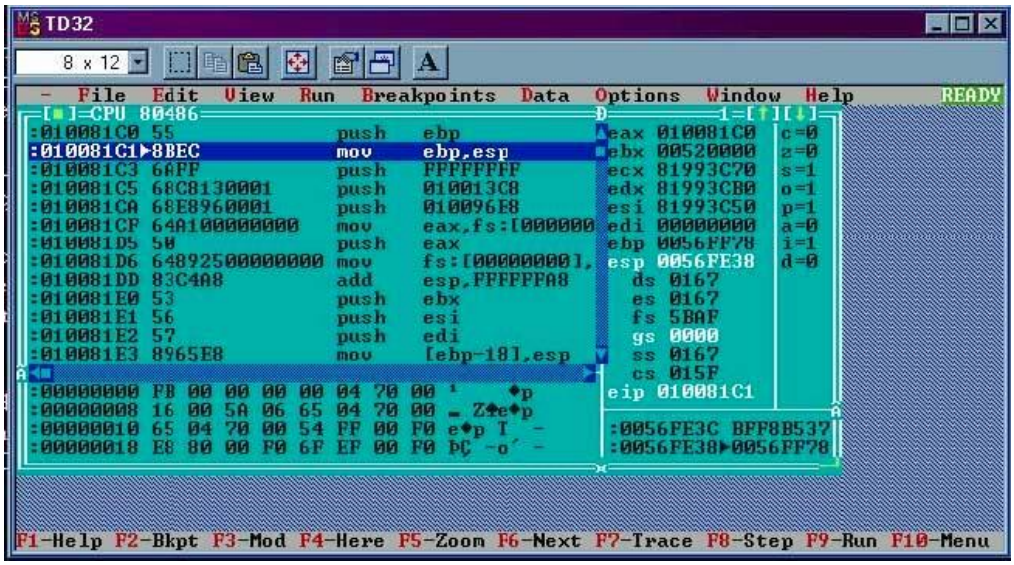
Programar: Se refiere a idear y ordenar las acciones que se realizarán en el marco de un proyecto; al anuncio de las partes que componen un acto o espectáculo; a la preparación de máquinas para que cumplan con una cierta tarea en un momento determinado; a la elaboración de programas para la resolución de problemas mediante ordenadores; y a la

preparación de los datos necesarios para obtener una solución de un problema a través de una calculadora electrónica, por ejemplo.

Programación de computadores: Un programa de computadora se puede definir como una secuencia de instrucciones que indica las acciones o tareas que han de ejecutarse para dar solución a un problema determinado.

Lenguaje Binario: Es aquel que numera empleando sólo ceros (0) y unos (1). Esto quiere decir que, en el marco de estos sistemas, cualquier cifra puede expresarse a partir de estos números. Este sistema es utilizado por las computadoras u ordenadores, que funcionan con un par de voltajes diferentes y que atribuyen el 0 al apagado y el 1 al encendido.

Lenguaje Assembler: Se define como un lenguaje de programación que se usa para dar directamente órdenes al ordenador. A diferencia de otros lenguajes, que usan el sistema operativo como intermediario para ejecutar las tareas (le dicen al sistema operativo que haga una cosa y este es quien se encarga de hacérselo saber al ordenador).



2. Punto- Clasificación los lenguajes de programación

1. Nivel de abstracción.

Según el nivel de abstracción, o sea, según el grado de cercanía a la máquina:

Lenguajes de bajo nivel: La programación se realiza teniendo muy en cuenta las características del procesador. Ejemplo: Lenguajes ensamblador.

Lenguajes de nivel medio: Permiten un mayor grado de abstracción pero al mismo tiempo mantienen algunas cualidades de los lenguajes de bajo nivel. Ejemplo: C puede realizar operaciones lógicas y de desplazamiento con bits, tratar todos los tipos de datos como lo que son en realidad a bajo nivel (números), etc.

Lenguajes de alto nivel: Más parecidos al lenguaje humano. Manejan conceptos, tipos de datos, etc., de una manera cercana al pensamiento humano ignorando (abstrayéndose) del funcionamiento de la máquina. Ejemplos: Java, Ruby.

Hay quien sólo considera lenguajes de bajo nivel y de alto nivel, (en ese caso, C es considerado de alto nivel).

2. Propósito.

Según el propósito, es decir, el tipo de problemas a tratar con ellos:

Lenguajes de propósito general: Aptos para todo tipo de tareas: Ejemplo: C.

Lenguajes de propósito específico: Hechos para un objetivo muy concreto. Ejemplo: Csound (para crear ficheros de audio).

Lenguajes de programación de sistemas: Diseñados para realizar sistemas operativos o drivers. Ejemplo: C.

Lenguajes de script: Para realizar tareas varias de control y auxiliares. Antiguamente eran los llamados lenguajes de procesamiento por lotes (batch) o JCL ("Job Control Languages"). Se subdividen en varias clases (de shell, de GUI, de programación web, etc.). Ejemplos: bash (shell), mIRC script, JavaScript (programación web).

3. Evolución histórica.

Con el paso del tiempo, se va incrementando el nivel de abstracción, pero en la práctica, los de una generación no terminan de sustituir a los de la anterior:

Lenguajes de primera generación (1GL): Código máquina.

Lenguajes de segunda generación (2GL): Lenguajes ensamblador.

Lenguajes de tercera generación (3GL): La mayoría de los lenguajes modernos, diseñados para facilitar la programación a los humanos. Ejemplos: C, Java.

Lenguajes de cuarta generación (4GL): Diseñados con un propósito concreto, o sea, para abordar un tipo concreto de problemas. Ejemplos: NATURAL, Matemática.

Lenguajes de quinta generación (5GL): La intención es que el programador establezca el qué problema ha de ser resuelto y las condiciones a reunir, y la máquina lo resuelve. Se usan en inteligencia artificial. Ejemplo: Prolog.

4. Manera de ejecutarse.

Según la manera de ejecutarse:

Lenguajes compilados: Un programa traductor traduce el código del programa (código fuente) en código máquina (código objeto). Otro programa, el enlazador, unirá los ficheros de código objeto del programa principal con los de las librerías para producir el programa ejecutable. Ejemplo: C.

Lenguajes interpretados: Un programa (intérprete), ejecuta las instrucciones del programa de manera directa. Ejemplo: Lisp.

También los hay mixtos, como Java, que primero pasan por una fase de compilación en la que el código fuente se transforma en "bytecode", y este "bytecode" puede ser ejecutado luego (interpretado) en ordenadores con distintas arquitecturas (procesadores) que tengan todos instalados la misma "máquina virtual" Java.

5. Manera de abordar la tarea a realizar.

Según la manera de abordar la tarea a realizar, pueden ser:

Lenguajes imperativos: Indican cómo hay que hacer la tarea, es decir, expresan los pasos a realizar. Ejemplo: C.

Lenguajes declarativos: Indican qué hay que hacer. Ejemplos: Lisp, Prolog. Otros ejemplos de lenguajes declarativos, pero que no son lenguajes de programación, son HTML (para describir páginas web) o SQL (para consultar bases de datos).

6. Paradigma de programación.

El paradigma de programación es el estilo de programación empleado. Algunos lenguajes soportan varios paradigmas, y otros sólo uno. Se puede decir que históricamente han ido apareciendo para facilitar la tarea de programar según el tipo de problema a abordar, o para facilitar el mantenimiento del software, o por otra cuestión similar, por lo que todos corresponden a lenguajes de alto nivel (o nivel medio), estando los lenguajes ensambladores "atados" a la arquitectura de su procesador correspondiente. Los principales son:

Lenguajes de programación procedural: Divide el problema en partes más pequeñas, que serán realizadas por subprogramas (subrutinas, funciones, procedimientos), que se llaman unas a otras para ser ejecutadas. Ejemplos: C, Pascal.

Lenguajes de programación orientada a objetos: Crean un sistema de clases y objetos siguiendo el ejemplo del mundo real, en el que unos objetos realizan acciones y se comunican con otros objetos. Ejemplos: C++, Java.

Lenguajes de programación funcional: La tarea se realiza evaluando funciones, (como en Matemáticas), de manera recursiva. Ejemplo: Lisp.

Lenguajes de programación lógica: La tarea a realizar se expresa empleando lógica formal matemática. Expresa qué computar. Ejemplo: Prolog.

Hay muchos paradigmas de programación: Programación genérica, programación reflexiva, programación orientada a procesos, etc.

7. Lugar de ejecución.

En sistemas distribuidos, según dónde se ejecute:

Lenguajes de servidor: Se ejecutan en el servidor. Ejemplo: PHP es el más utilizado en servidores web.

Lenguajes de cliente: Se ejecutan en el cliente. Ejemplo: JavaScript en navegadores web.

8. Concurrencia.

Según admitan o no concurrencia de procesos, esto es, la ejecución simultánea de varios procesos lanzados por el programa:

Lenguajes concurrentes. Ejemplo: Ada.

Lenguajes no concurrentes. Ejemplo: C.

9. Interactividad.

Según la interactividad del programa con el usuario u otros programas:

Lenguajes orientados a sucesos: El flujo del programa es controlado por la interacción con el usuario o por mensajes de otros programas/sistema operativo, como editores de texto, interfaces gráficos de usuario (GUI) o kernels. Ejemplo: VisualBasic, lenguajes de programación declarativos.

Lenguajes no orientados a sucesos: El flujo del programa no depende de sucesos exteriores, sino que se conoce de antemano, siendo los procesos batch el ejemplo más claro (actualizaciones de bases de datos, colas de impresión de documentos, etc.). Ejemplos: Lenguajes de programación imperativos.

10. Realización visual.

Según la realización visual o no del programa:

Lenguajes de programación visual: El programa se realiza moviendo bloques de construcción de programas (objetos visuales) en un interfaz adecuado para ello. No

confundir con entornos de programación visual, como Microsoft Visual Studio y sus lenguajes de programación textuales (como Visual C#). Ejemplo: Mindscript.

Lenguajes de programación textual: El código del programa se realiza escribiéndolo. Ejemplos: C, Java, Lisp.

11. Determinismo.

Según se pueda predecir o no el siguiente estado del programa a partir del estado actual:

Lenguajes deterministas. Ejemplos: Todos los anteriores.

Lenguajes probabilísticos o no deterministas: Sirven para explorar grandes espacios de búsqueda, (como gramáticas), y en la investigación teórica de hipercomputación. Ejemplo: mutt (generador de texto aleatorio).

12. Productividad.

Según se caractericen por tener virtudes útiles o productivas, u oscuras y enrevesadas:

Lenguajes útiles o productivos: Sus virtudes en cuanto a eficiencia, sencillez, claridad, productividad, etc., motiva que sean utilizados en empresas, administraciones públicas y/o en la enseñanza. Ejemplos: Cualquier lenguaje de uso habitual (C, Java, C++, Lisp, Python, Ruby,…).

Lenguajes esotéricos o exóticos: Inventados con la intención de ser los más raros, oscuros, difíciles, simples y/o retorcidos de los lenguajes, para diversión y entretenimiento de frikis programadores. A veces exploran nuevas ideas en programación. Ejemplo: Brainfuck.

POR MEDIO DE ESTE LINK, ECONTRARAS MAS INFORMACION ACERCA DE LOS LENGUAJES DE PROGRAMACION

http://www.larevistainformatica.com/clasificacion-de-los-lenguajes-de-programacion.html

Presentación en diapositivas de los lenguajes de alto, medio y bajo:

http://es.slideshare.net/ClariitaPM/compiladores-imterpretadores-lenguaje-de-alto-medio-y-bajo-nivel-y-lenguaje-c#btnNext

4. PUNTO -VENTAJAS Y DESVENTAJAS DE LOS LENGUAJES DE ALTO, MEDIO Y BAJO NIVEL

LENGUAJES	VENTAJAS	DESVENTAJAS
ALTO NIVEL	-Genera un código más sencillo y comprensible. -escribir un código valido para diversas páginas y, posiblemente, sistemas operativos.	-Reducción de velocidad al ceder al trabajo del bajo nivel de la máquina. -Algunos requieren que la maquina cliente posea una determinada plataforma.
MEDIO NIVEL	-Permiten un mayor grado de abstracción -Son precisos para ciertas aplicaciones como la creación de sistemas operativos.	-No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación.
BAJO NIVEL	-Mayor adaptación al equipo. -Posibilidad de obtener la máxima velocidad con mínimo uso de memoria.	-Imposibilidad de escribir código independiente de la máquina. -Mayor dificultad en la programación y en la comprensión de los programas. -El programador debe conocer más de un centenar de instrucciones. -Es necesario conocer en detalle la arquitectura de la máquina.

5. Punto - Palabras reservadas en programación

C#

Las palabras reservadas son identificadores reservados predefinidos que tienen un significado especial y no se pueden utilizar como identificadores en sus programas, excepto si llevan delante el carácter @ como prefijo. Así, por ejemplo @for es un identificador

válido, pero no for ya que es una palabra reservada.

C++

Las palabras reservadas son identificadores predefinidos reservados que tienen significados especiales y no se pueden utilizar como identificadores de sus programas. Los nombres con subrayados a la izquierda son extensiones de Microsoft.

Java

Estas 48 palabras están definidas en el lenguaje Java. Estas palabras reservadas, combinadas con la sintaxis de los operadores y separadores, forman la definición del lenguaje Java. Estas palabras reservadas no se pueden utilizar como nombres en sus programas Java en variables, clases o métodos. True, false, y null no son palabras clave, pero tienen el significado de palabras reservadas y tampoco se pueden utilizar como nombres en sus programas.

http://www.mhe.es/universidad/informatica/8448198441/archivos/apendice_general_3.pdf

Este link es la fuente de la cual se extrajo la información, en ella podrás encontrar contenido adicional al que se está presentando.

6. Punto- la sintaxis de un lenguaje de programación

La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C.

```
<style type="text/css">
.appearance_page_custom-header #heading {
        border: none;
}
#heading h1,
#desc {
        font-family: "Helvetica Neue", Arial, Helvetica, "Nimbus Sans L", sans-serif;
}
#heading h1 {
        margin: 0;
}
#heading h1 a {
        font-size: 32px;
        line-height: 36px;
        text-decoration: none;
}
#desc {
        font-size: 14px;
        line-height: 23px;
        padding: 0 0 3em;
}
<?php
        // If the user has set a custom color for the text use that
        if ( get_header_textcolor() != HEADER_TEXTCOLOR ) :
```

Sintaxis y código

http://librosweb.es/javascript/capitulo_1/sintaxis.html

7-Punto- Lenguajes de quinta generación

- Un lenguaje de programación de quinta generación es un lenguaje de programación basado en la resolución de problemas utilizando restricciones dadas al programa, en lugar de utilizar un algoritmo escrito por un programador. Más restricción basada y lenguajes de programación lógica y algunos lenguajes declarativos son lenguajes de quinta generación.

Mientras que los lenguajes de programación de cuarta generación están diseñados para desarrollar programas específicos, lenguajes de quinta generación están diseñados para hacer que el equipo a resolver un problema dado sin que el programador. De esta manera, el programador sólo tiene que preocuparse de lo que es necesario resolver y qué condiciones deben cumplirse, sin tener que preocuparse acerca de cómo implementar una rutina o algoritmo para resolver los problemas. Lenguajes de quinta generación se utilizan principalmente en la investigación de la inteligencia artificial. Prolog, OPS5, y el mercurio son ejemplos de lenguajes de quinta generación.

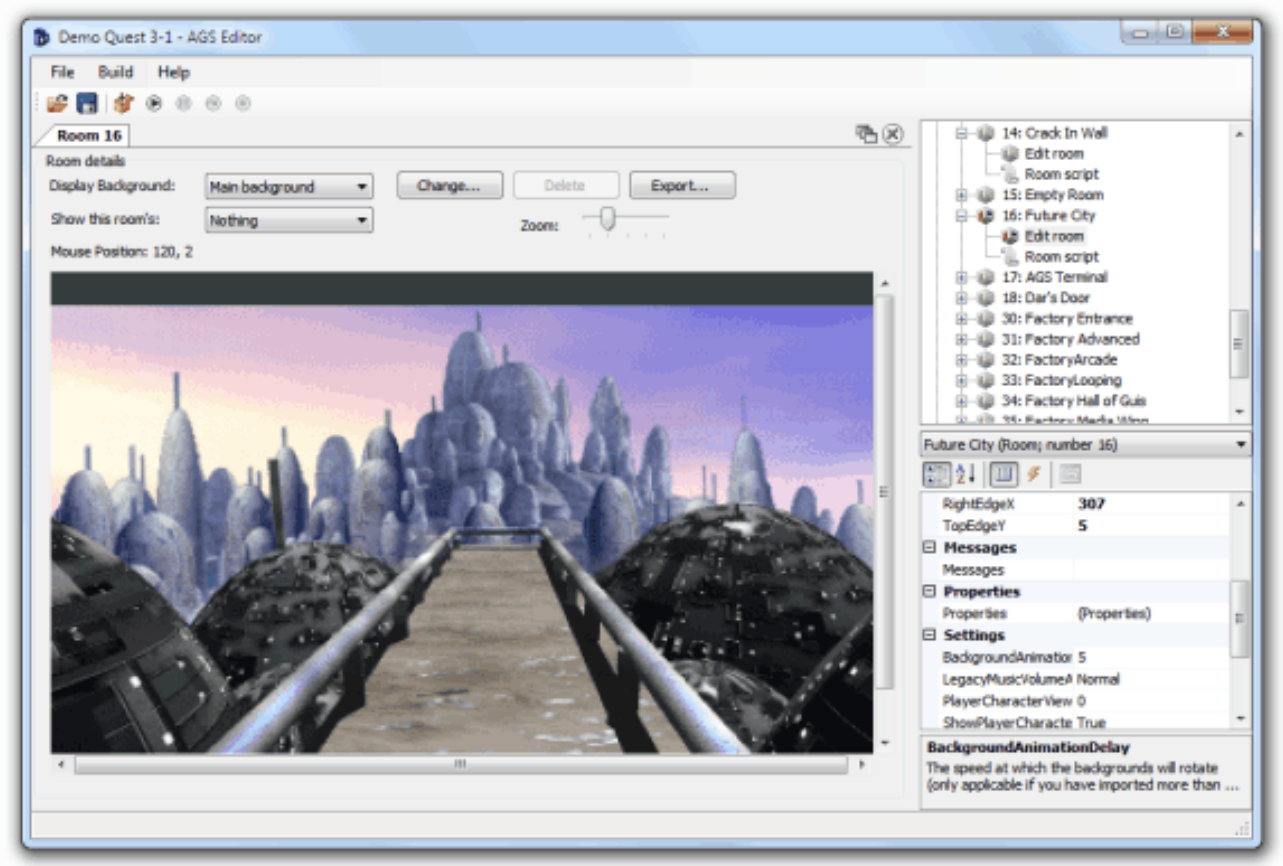
http://es.scribd.com/doc/136096003/Lenguajes-de-quinta-Generacion-Inteligencia-Artificial
http://centrodeartigos.com/articulos-educativos/article_10412.html

8. Punto- Programas que permita la creación de juegos

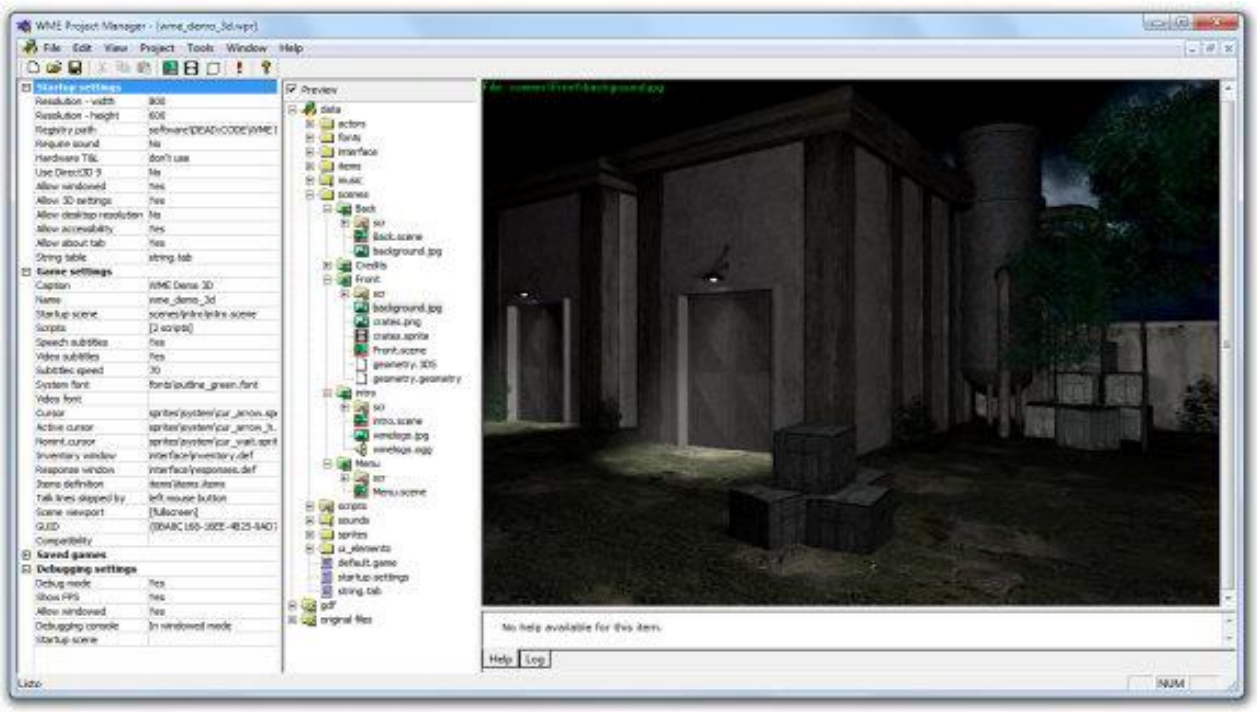
Programas que permiten la creación de juegos	VENTAJAS	DESVENTAJAS
ADVENTURE GAME	- Permite crear tu propia aventura - Resultados impresionantes - Potente y efectivo	- Los efectos especiales hay que programarlos, y hay que saber inglés para entender los manuales - Hay que usar un programa de dibujo
WinterMute Engine	Este es diseñador de graficas de aventuras con resoluciones muy alto y además de que ofrece mucha flexibilidad a todo el entorno de los personajes que se va a diseñar. Las resoluciones pueden alcanzar hasta la vista 1024x768, aceleración 3D, un amplio soporte multimedia, scripting muy similar a PHP o C++, etc. El proyecto es GNU, esto quiere decir que es de libre uso	No tiene desventajas
Visionaire	Brinda la libertad de crear aventuras gráficas con un rendimiento totalmente profesional. Permite utilizar OpenGL, amplio soporte multimedia, grandes resoluciones de 1280x1024, entre otras características.	No tiene desventajas
Alpaca	- Es el motor perfecto para todos aquellos que estén familiarizados con Flash, ya que está preparado para diseñar aventuras gráficas con esta tecnología. -Posee ventajas de tecnología flash	-Tiene desventajas de la tecnología flash
	- Este se basa en el clásico	No tiene

Sludge	Sludge, nace OpenSludge, un motor de desarrollo de aventuras gráficas de código abierto y multiplataforma. - Cuenta con una gran cantidad de características, disponibles para Windows, Linux y Mac OS.	

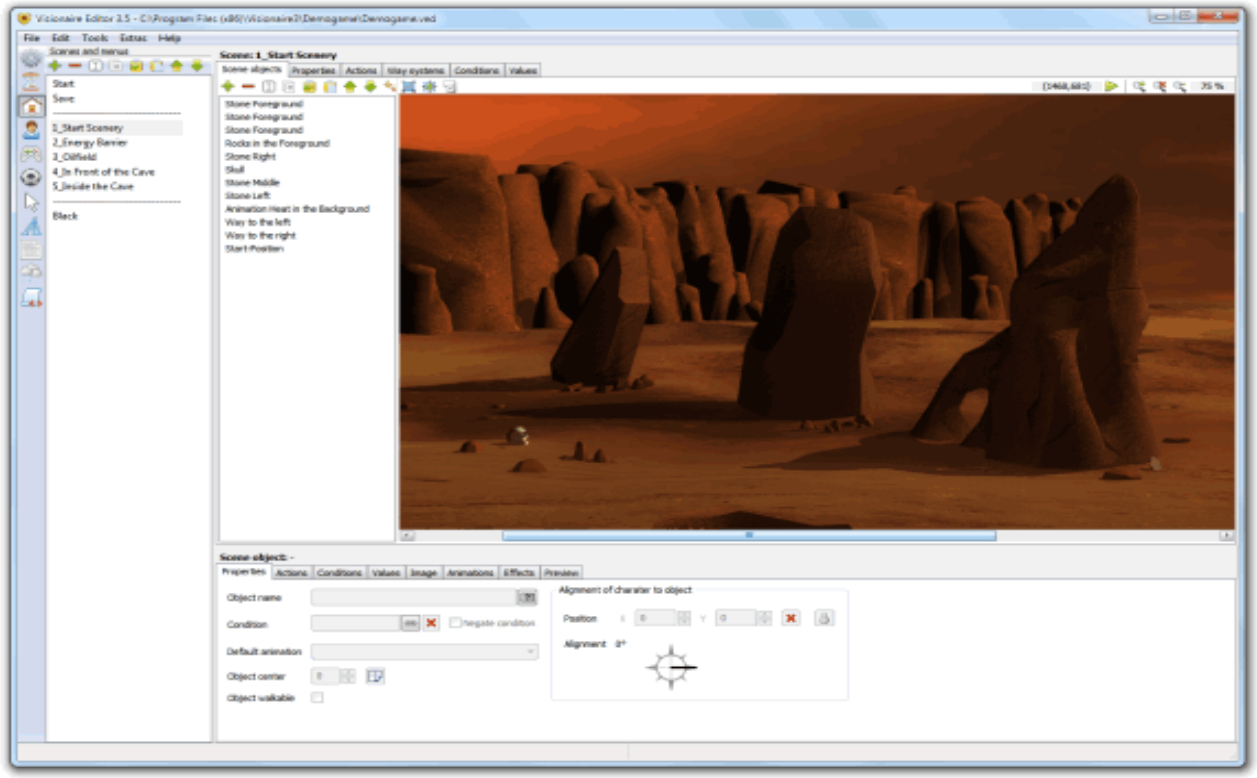
1. ADVENTURE GAME



2. WinterMute Engine

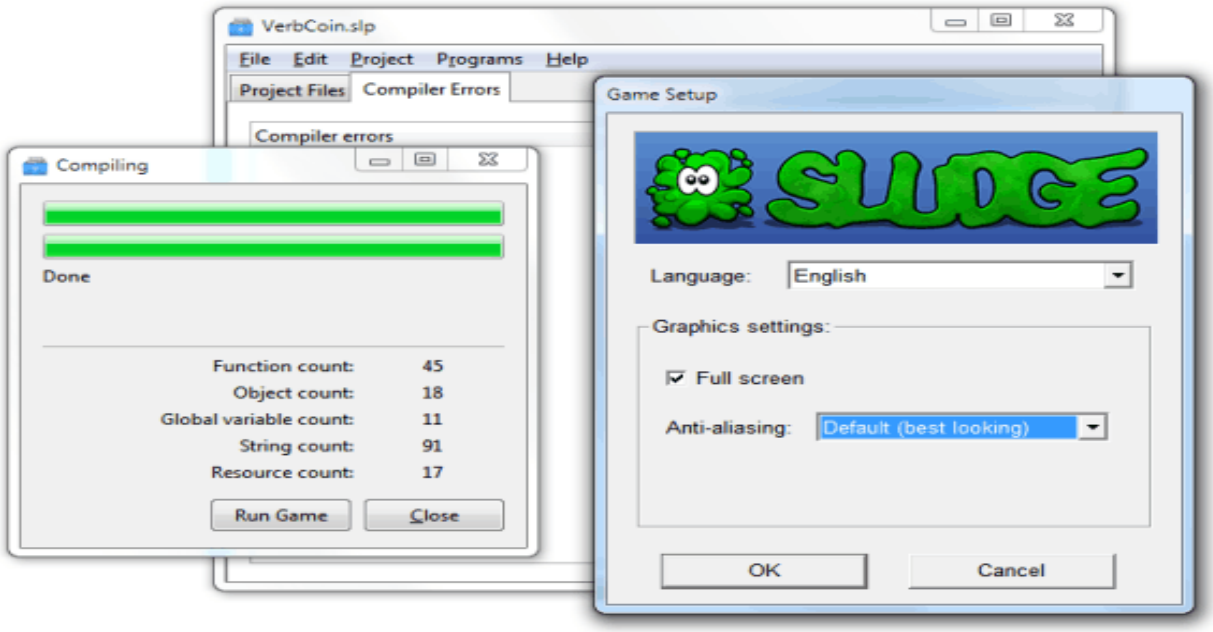


3. Visionaire



4. Alpaca

5. Sludge



Conclusión:

Para entender los lenguajes de programación primero debemos tener claros ciertos conceptos, los cuales nos ayudaran a maximizar nuestro léxico y además de esto entender de una manera mucho mejor estos lenguajes, como pudimos ver en el taller existen cantidad de lenguajes pero en los que nos enfocamos son en los lenguajes de alto, medio y bajo nivel, miramos sus ventajas y desventajas, estos lenguajes nos ayudan a una mejor adaptación del equipo, posibilidad de obtener la máxima velocidad con mínimo uso de memoria.

Cibergrafia:

http://www.desarrolloweb.com/articulos/2358.php

http://es.scribd.com/doc/136096003/Lenguajes-de-quinta-Generacion-Inteligencia-Artificial

http://centrodeartigos.com/articulos-educativos/article_10412.html

http://es.slideshare.net/ClariitaPM/compiladores-imterpretadores-lenguaje-de-alto-medio-y-bajo-nivel-y-lenguaje-c#btnNext

http://www.larevistainformatica.com/clasificacion-de-los-lenguajes-de-programacion.html