



INSTITUTO TECNOLÓGICO NACIONAL DE MÉXICO EN CELAYA

Materia: Lenguajes y Autómatas II
ACTIVIDAD 3

Profesor: ISC. Ricardo González
González

Alumnos:

Isaac Salvador Bravo Estrada
2003048

Guillermo Peasland Aguilar 20030737
María del Carmen Chávez Patiño
20030296

Luis Fernando Mendoza Javalera
19030536

Fecha de Entrega: 03 de Septiembre
de 2024

EQUIPO NO. 3



DEPARTAMENTO DE SISTEMAS COMPUTACIONALES E INFORMÁTICA

ASUNTO: **SOLICITUD DE ACTIVIDADES**

Celaya, Guanajuato, 27 / agosto / 2024

LENGUAJES Y AUTÓMATAS II

DOCENTE DESIGNADO: ISC. RICARDO GONZÁLEZ GONZÁLEZ
SEMESTRE AGOSTO-DICIEMBRE 2024

ACTIVIDAD 3 (VALOR 44 PUNTOS)

LEA CUIDADOSAMENTE, Y REALICE LAS SIGUIENTE ACTIVIDADES, CONSIDERANDO LOS CRITERIOS DE CALIDAD PROPUESTOS EN LOS DOCUMENTOS DE LA [GUÍA TUTORIAL](#), Y LA [RÚBRICA DE EVALUACIÓN](#).

EL LECTOR DEBE TOMAR MUY EN CUENTA QUE ESTA ACTIVIDAD ES UN EXAMEN, Y NO UNA SIMPLE TAREA, PUES DEMANDA DEDICACIÓN PARA INVESTIGAR, LEER, ANALIZAR, REDACTAR, ILUSTRAR Y PROPONER DE MANERA PROFESIONAL LOS TEMAS PROPUESTOS EN LA ESTRUCTURA TEMÁTICA DE ESTA ASIGNATURA.

1. INTRODUCCIÓN AL DISEÑO DE LENGUAJE DE PROGRAMACIÓN.

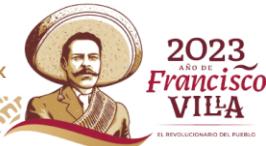
CONTEXTO : ESTA ACTIVIDAD DEMANDARÁ DE USTED Y DE SU EQUIPO PONER A TRABAJAR MUCHAS DE LAS HABILIDADES QUE SEGURAMENTE YA HA ADQUIRIDO ANTES.

POR ELLO SE LE RECOMIENDA QUE ADMINISTRE CUIDADOSAMENTE SU TIEMPO Y SIGA LAS INSTRUCCIONES RIGUROSAMENTE.

EL MATERIAL QUE SE PROPORCIONA REFIERE A UN TEMA MUY IMPORTANTE PARA ESTA ASIGNATURA, Y ES EL "DISEÑO DE LENGUAJES DE PROGRAMACIÓN", LA PARTE MEDULAR DE ÉSTA EVALUACIÓN.



Av. Antonio García Cubas #600 esq. Av. Tecnológico, Colonia Alfredo V. Bonfil, C.P. 38010
Celaya, Gto. Tel. 01 (461) 611 75 75 e-mail: lince@celaya.tecnm.mx tecnm.mx | celaya.tecnm.mx





ESTE MATERIAL PERTENECE A LOS SIGUIENTES AUTORES Y SE ENCUENTRA EN INTERNET BAJO EL SIGUIENTE TARJETA BIBLIOGRÁFICA.

"INTÉRPRETES Y DISEÑO DE LENGUAJES DE PROGRAMACIÓN"

José Emilio Labra Gayo Juan
Manuel Cueva Lovelle
Raúl Izquierdo Castanedo
Aquilino Adolfo Juan Fuente
M^a Cándida Luengo Díez Francisco
Ortín Soler

Profesores de la Universidad de Oviedo Apartado de correos 1910, 33080 - Oviedo (España). Teléfono +34609108131.

Editorial: SERVITEC ISBN: 84-688-4210-9

ES UN MATERIAL MUY INTERESANTE QUE ABORDA DE MANERA CLARA Y CONCRETA CONCEPTOS IMPORTANTES A CONSIDERAR EN EL DISEÑO DE LENGUAJES DE PROGRAMACIÓN Y A LA VEZ SE APOYA DE EJEMPLOS E ILUSTRACIONES QUE COMPLEMENTAN LA COMPRENSIÓN DE LOS TEMAS.

USAREMOS ESTE MATERIAL CON FINES ACADÉMICOS ÚNICAMENTE Y ACCEDEREMOS AL CONOCIMIENTO DE LOS TEMAS PROPUESTOS DE UN MODO ANALÍTICO.

ESTA ACTIVIDAD NO PRETENDE QUE EL ESTUDIANTE APRENDA TODOS LOS CONCEPTOS MEMORIZANDO, SINO QUE ABORDE LOS FUNDAMENTOS PARA EL DISEÑO DE LENGUAJES FORMALES, CONCRETAMENTE EN EL ÁMBITO DE LA PROGRAMACIÓN.

POR ÚLTIMO, ANTES DE EXPLICAR LO QUE SE ESPERA COMO PRODUCTOS DE EVIDENCIA, EL ESTUDIANTE DEBERÁ TOMAR CONCIENCIA DE LA IMPORTANCIA DE ESTE MATERIAL, Y DE SU CORRECTO ANÁLISIS, PUES SE DEBERÁ DEDICAR A ESTO POR VARIOS DÍAS Y SU PESO EN PUNTOS ES IMPORTANTE PARA LA ACREDITACIÓN DE ESTA PRIMERA EVALUACIÓN.

1. DESCARGUE Y LEA DETENIDAMENTE EL DOCUMENTO QUE SE ADJUNTA. HAGA USO DE SU CAPACIDAD DE ANÁLISIS Y DESPUÉS DE UNA LECTURA PRELIMINAR, HAGA OTRA LECTURA DE SELECCIÓN EN LA CUAL SUBRAYE DE FORMA FÍSICA O DIGITAL AQUELLO QUE ES LO MÁS RELEVANTE A CONSIDERAR.

NO OMITA PIEZAS DE INFORMACIÓN IMPORTANTES, SOLO POR HACER EL TRABAJO MÁS RÁPIDO, PUES CON ELLO SOLO LOGRARÁ HACER UN TRABAJO DE POBRE CALIDAD.

2. EL DOCUMENTO CONSTA DE 8 SECCIONES DESDE LA 2.1 HASTA LA 2.8, Y PARA CADA SECCIÓN USTED DEBERÁ GENERAR UN PRODUCTO DE EVIDENCIA QUE DEBERÁ SER ALGUNO DE LOS SIGUIENTES :
 - a. UN VIDEO COLOCADO EN YOUTUBE O EN DRIVE : EN ESTE VIDEO HECHO POR USTED Y/O EQUIPO Y CON LA CÁMARA QUE TENGA DISPONIBLE, DEBERÁ EXPLICAR DETALLADAMENTE DE QUE TRATA LA SECCIÓN Y LO MÁS IMPORTANTE QUÉ EXPLICA DICHA SECCIÓN. ESTO SERÍA SIMILAR A UNA EXPOSICIÓN FRENTA A GRUPO.





- b. UN RESUMEN POR ESCRITO A MANO Y DETALLADO SOBRE LA SECCIÓN QUE USTED DECIDA.
 - c. UN MAPA CONCEPTUAL MUY COMPLETO, USANDO COLORES Y DE CREATIVIDAD PROFESIONAL SOBRE LA SECCIÓN QUE USTED DECIDA.
 - d. UN CUADRO COMPARATIVO DETALLADO Y BIEN ESTRUCTURADO DONDE SE SINTETICE CORRECTAMENTE LO EXPUESTO EN LA SECCIÓN QUE USTED DECIDA.
ESTAS CUATRO FORMAS DE GENERAR EVIDENCIAS DE APRENDIZAJE SE DEBERÁN REPETIR EN DOS OCASIONES, ESTO ES, GENERARÁ DOS VIDEOS, DOS RESÚMENES, DOS MAPAS MENTALES Y DOS CUADROS COMPARATIVOS.
3. ¿ QUÉ SE CALIFICARÁ ?

LA RÚBRICA PARA EVALUAR ESTA ACTIVIDAD ESTARÁ INTEGRADA POR LOS SIGUIENTES CRITERIOS, MISMOS QUE EN LA PLATAFORMA DE CLASSROOM TAMBIÉN PODRÁ CONSULTAR..

- a. **LA OPORTUNIDAD.** SI EL TRABAJO FUE ENTREGADO OPORTUNAMENTE.
- b. **LA COMPRENSIÓN.** SE VALORARÁ EL GRADO DE COMPRENSIÓN DEL TEXTO ANALIZADO.
- c. **LA CALIDAD.** SI LAS EVIDENCIAS ENVIADAS CORRESPONDEN A LA CALIDAD ESPERADA PARA ESTE NIVEL PROFESIONAL QUE SE CURSA.
- d. **LA CAPACIDAD DE SÍNTESIS.** SI LAS EVIDENCIAS ENTREGADAS CONDENSAN LA ESENCIA DE LA SECCIÓN ABORDADA, O EN SU DEFECTO SE OMITIERON CONCEPTOS CON EL AFÁN DE SÓLO ENTREGAR UNA CARENTE EVIDENCIA.
- e. **LA CREATIVIDAD.** LA MANERA EN QUE SE EXPRESAN LOS CONCEPTOS Y EL TRATAMIENTO QUE SE DA A LA INFORMACIÓN ANALIZADA PARA QUE ÉSTA SEA COMPRESIBLE EN SU ESENCIA.

IMPORTANTE : CUENTA CON EL TIEMPO SUFFICIENTE PARA REALIZAR ESTA ACTIVIDAD Y SUMAR PUNTOS IMPORTANTES A SU CALIFICACIÓN DE LA EVALUACIÓN 1.

IMPORTANTE : PARA LA ENTREGA DE LOS VIDEOS, BASTARÁ QUE INCLUYA LA LIGA QUE CORRESPONDA A ÉSTOS PRODUCTOS CARGADOS EN LA PLATAFORMA DE YOUTUBE.

PUEDE USAR OTRA PLATAFORMA Y TECNOLOGÍAS PARA GENERAR Y ALMACENAR SUS VIDEOS, COMO YA ANTES LO HA HECHO A TRAVÉS DEL USO DE GOOGLE DRIVE DE SU CUENTA INSTITUCIONAL.

IMPORTANTE : TODO EL MATERIAL ESCRITO DEBERÁ SER HECHO A MANO Y BASADO EN LOS CRITERIOS ANTES SEÑALADOS EN LA DOCUMENTACIÓN DE INICIO DE SEMESTRE.





CONSIDERACIONES.

CADA UNO DE LOS PUNTOS ANTERIORES DEBE SER DESARROLLADO CON LA PROFUNDIDAD ACORDE A UN NIVEL PROFESIONAL, Y APEGÁNDOSE COMPLETAMENTE A LAS DIRECTRICES DE LA GUÍA TUTORIAL.

NO CONCIBA ESTE TRABAJO, COMO UN SIMPLE RESUMEN O EJERCICIO DE TRANSCRIPCIÓN, PUES EL VALOR INDICADO AL INICIO DE ESTA ACTIVIDAD LE DARÁ A USTED UNA BUENA IDEA DE LO QUE SE ESPERA DE ELLA, EN CUANTO A CALIDAD Y EL APRENDIZAJE OBTENIDO, MISMO QUE SERÁ PUESTO A PRUEBA MEDIANTE UN EXAMEN ESCRITO O BIEN ORAL EN CLASE.

SI DECIDIÓ ELABORAR ESTA ACTIVIDAD EN EQUIPO, CADA INTEGRANTE DE ÉSTE DEBERÁ POSEER EL MISMO NIVEL DE CONOCIMIENTO, PUES TAN SOLO REPARTIR TEMAS ENTRE LOS INTEGRANTES DEL EQUIPO, SUPONDRIÁ UN GRAVE ERROR DE INTERPRETACIÓN A LA INTENCIÓN DIDÁCTICA REAL DE ESTA ACTIVIDAD.

POR ÚLTIMO, ESTA ACTIVIDAD SOLO SE PODRÁ DESARROLLAR EN EQUIPO, SI SE REGISTRÓ EN UNO PREVIAMENTE, UTILIZANDO EL FORMATO ENTREGADO EN LA ACTIVIDAD INICIAL. DE LO CONTRARIO DEBERÁ ELABORAR Y ENTREGAR LA ACTIVIDAD DE FORMA INDIVIDUAL.

LA ENTREGA DE DICHO REGISTRO SE HARÁ VÍA CORREO ELECTRÓNICO ENVIANDO ÉSTE AL PROFESOR DESIGNADO, Y POSTERIORMENTE EN CLASE ENTREGANDO LA HOJA EN FÍSICO.

OBSERVACIONES:

- CADA HOJA QUE ENTREGUE DE SU ACTIVIDAD, DEBERÁ ESTAR FIRMADA AL MARGEN DERECHO, INCLUIDA LA PROPIA SOLICITUD DE LA ACTIVIDAD.
- INTEGRE TODO SU TRABAJO EN UN SOLO ARCHIVO DE TIPO .PDF, Y ASIGNE EL NOMBRE QUE A CONTINUACIÓN SE INDICA.

NO OLVIDE ANEXAR LAS HOJAS DE ESTA ACTIVIDAD Y DE SU TRABAJO DESPUÉS DE SU PORTADA.

- UNA VEZ ELABORADA SU ACTIVIDAD, RECUERDE DIGITALIZARLA Y NOMBRARLA EN BASE A LA NOMENCLATURA QUE SE INDICA MÁS ADELANTE EN ESTE DOCUMENTO.
- SI SUS EVIDENCIAS ENVIADAS POR CORREO, NO CUMPLEN CON LA NOMENCLATURA SOLICITADA, NO SERÁN CONSIDERADAS COMO EVIDENCIAS PARA SU EVALUACIÓN.
- POR ÚLTIMO, POR FAVOR GESTIONE APROPIADAMENTE SU TIEMPO, Y SEA PUNTUAL EN SU ENTREGA Y ASÍ EVITAR PROBLEMAS DE NULIDAD POR EXTEMPORANEIDAD.





LA NOMENCLATURA SOLICITADA PARA ENVIAR SU TRABAJO ES LA SIGUIENTE :

AAAA-MM-DD_TNM_CELAYA_MATERIA_DOCUMENTO_[EQUIPO]_NOCTROL_APELLIDOS_NOMBRE_SEM.PDF

(NOTA : * TODO DEBE SER ESCRITO USANDO LETRAS MAYÚSCULAS ***)**

DONDE :

TNM_CELAYA	: INSTITUCIÓN ACADÉMICA
AAAA	: AÑO
MM	: MES
DD	: DÍA
MATERIA	: LAI , LI MÁS EL GRUPO (-A, -B, -C)
DOCUMENTO	: AI-ACTIVIDAD 1, P1-PRACTICA 1, R1-REPORTE 1, TI-TAREA 1, PG1-PROGRAMA, ETC. (CAMBIANDO EL NÚMERO CONSECUТИVO POR EL QUE CORRESPONDA)
[EQUIPO]	: NÚMERO DEL EQUIPO QUE CORRESPONDA SEGÚN INDICACIÓN DEL PROFESOR. [OPCIONAL]
NOCTROL	: SU NÚMERO DE CONTROL
APELLIDOS	: SUS APELLIDOS
NOMBRE	: SU NOMBRE
SEM	: EL PERÍODO SEMESTRAL EN CURSO: AGO-DIC

EJEMPLO :

SI EL TRABAJO SE SOLICITÓ EN EQUIPO.

2024-08-27_TNM_CELAYA_LAI-A_A3_EQUIPO_99_9999999_PEREZ_PEREZ_JUAN_AGO-DIC24.PDF

DONDE EL NOMBRE DEBERÁ CORRESPONDER AL JEFE DE EQUIPO QUE HACE LA ENTREGA DEL TRABAJO.

SI EL TRABAJO SE SOLICITÓ INDIVIDUALMENTE.

2024-08-27_TNM_CELAYA_LAI-A_A3_9999999_PEREZ_PEREZ_JUAN_AGO-DIC24.PDF





FECHA Y HORA DE ENTREGA:

LA INDICADA EN LA PLATAFORMA VIRTUAL.

EN CASO DE QUE EL TRABAJO SE HAYA SOLICITADO EN EQUIPO, EL JEFE DEL MISMO SERÁ EL ÚNICO RESPONSABLE DE ENVIAR LA ACTIVIDAD EN LA PLATAFORMA VIRTUAL.

MUY IMPORTANTE:

1. DESPUÉS DE LA HORA INDICADA EN LA PLATAFORMA VIRTUAL (AÚN CUANDO SOLO SEA UN MINUTO O VARIOS), LA ACTIVIDAD SERÁ CONSIDERADA COMO EXTEMPORÁNEA Y NO CONTARÁ COMO EVIDENCIA PARA SU EVALUACIÓN.

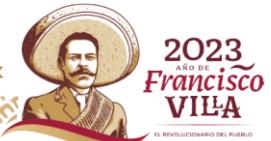
SE LE SUGIERE ENVIAR CON ANTICIPACIÓN SU ACTIVIDAD A FIN DE EVITAR CONFLICTOS POR NO ENTREGAR ÉSTA A TIEMPO.

BAJO NINGÚN PRETEXTO O JUSTIFICACIÓN SE ACEPTARÁN LOS TRABAJOS EXTEMPORÁNEOS, EVITE LA PENA DE RECORDAR A USTED QUE EL VALOR DE LA PUNTUALIDAD ES PARTE IMPORTANTE DE SUS EVIDENCIAS Y ES EL PRIMER PUNTO QUE SE HA DE EVALUAR.

2. NO OLVIDE ANEXAR A SU ARCHIVO .PDF DE EVIDENCIAS UNA PORTADA PROFESIONAL, Y ESTA SOLICITUD DE ACTIVIDADES CON TODAS LAS HOJAS FIRMADAS EN EL MARGEN DERECHO.
3. POR ÚLTIMO, TODA EVIDENCIA GENERADA QUE CONTENGA AL MENOS UNA TRANSCRIPCIÓN DE CUALQUIER FUENTE Y DE CUALQUIER TIPO, ES DECIR CON MATERIAL PLAGIADO SERÁ ANULADA DE FORMA INCONTROVERTIBLE.



Av. Antonio García Cubas #600 esq. Av. Tecnológico, Colonia Alfredo V. Bonfil, C.P. 38010
Celaya, Gto. Tel. 01 (461) 611 75 75 e-mail: lince@celaya.tecnm.mx tecnm.mx | celaya.tecnm.mx



2.1 ASPECTOS LINGÜÍSTICOS

Los lenguajes de programación son herramientas diseñadas para construir programas que deben ser comprendidos tanto por seres humanos como por computadoras. A diferencia del lenguaje natural, que se desarrolla y se comprende a través de la costumbre y la experiencia, los lenguajes de programación son definidos por una autoridad, como el diseñador del lenguaje o un comité especializado.

La comunicación efectiva en programación requiere una comprensión mutua entre humanos y máquinas, lo cual es complejo debido a la naturaleza distinta de ambos. Mientras que los seres humanos interpretan los símbolos basados en experiencia y contexto, las máquinas requieren una definición precisa y estructurada para ejecutar tareas.

Por eso, los lenguajes de programación son un compromiso entre la legibilidad para los humanos y la traducibilidad para las máquinas.

El estudio de los lenguajes de programación puede abordarse desde tres perspectivas clásicas de la Semiótica, adaptadas por C. Morris en 1938

Sintaxis: se refiere al formato y escritura.

Semántica: Estudia el comportamiento cuando se ejecuta.

Pragmática: Examina las técnicas y Métodos.

2.2: PRINCIPIOS DE DISEÑO

El diseño de lenguajes de programación implica tomar buenas decisiones sobre qué características incluir y cómo balancear diferentes necesidades. Un buen diseño busca optimizar la utilidad y eficiencia al igual que eficacia del lenguaje sin sobrecargar al programador con complejidad innecesaria. Algunos principios clave del diseño de lenguaje son:

CONCISIÓN NOTACIONAL: El lenguaje debe ser lo suficientemente claro y simple para que los programadores puedan pensar y expresar algoritmos sin excesiva complejidad, pero detallado lo suficiente para ser útil.

ORTOGONALIDAD: Las características del lenguaje deben ser lo más independientes posibles entre sí, permitiendo su combinación sin generar excepciones o incoherencias.

ABSTRACCIÓN: Un buen lenguaje de programación permite al programador evitar la repetición innecesaria de código.

SEGURIDAD: Para mejorar la fiabilidad del software, los lenguajes deben ser capaces de rechazar programas incorrectos, idealmente desde la etapa de compilación.

EXPRESIVIDAD: El lenguaje debe permitir al programador expresar sus intenciones de manera efectiva. Sin embargo, demasiada expresividad puede comprometer la seguridad del software, por lo que algunos lenguajes limitan la expresividad.

EXTENSIBILIDAD: Un lenguaje de programación debe ofrecer mecanismos para que los programadores puedan extender su funcionalidad mediante la creación de nuevas construcciones.

PORTABILIDAD: Es esencial que los programas escritos en un lenguaje sean ejecutables en diferentes entornos computacionales.

EFICIENCIA: El lenguaje debe permitir la escritura de algoritmos eficientes y ofrecer técnicas de optimización que mejoren el rendimiento de los programas.

LIBRERIAS E INTERACCIÓN CON EL EXTERIOR: La disponibilidad de un conjunto de librerías que facilite el desarrollo rápido de aplicaciones es crucial para la popularidad del lenguaje.

ENTORNO: Aunque el entorno de desarrollo no es parte intrínseca del lenguaje, un entorno de desarrollo potente y bien documentado puede ser determinante para la adopción y éxito de un lenguaje. La disponibilidad de documentación, ejemplos y una comunidad activa también contribuyen a su aceptación.



2.3 Definición de un lenguaje.

Aspecto	Descripción
Definición del Lenguaje	Una definición completa y precisa del lenguaje que permite desarrollar implementaciones en diferentes entornos y sistemas. Los programas deben ser procesables por cualquier implementación del lenguaje.
Proceso de Estandarización	Surge para asegurar la unicidad del lenguaje en diferentes plataformas. Involucra la creación de un estándar formal de la sintaxis y semántica del lenguaje.
Autoridad del Estándar	Puede ser un diseñador individual o una agencia de estandarización (como ANSI, ISO, ECMA). Usualmente un comité de personas de diversos orígenes colabora en la creación del estándar.
Primera Versión del Estándar	Limpia ambigüedades, corrige defectos, y define un lenguaje más portable. Los implementadores deben ajustar sus implementaciones al estándar.
Desviaciones del Estándar	<ul style="list-style-type: none">• Extensiones: Añaden características sin romper la compatibilidad.• Modificaciones: Alteran características del estándar, pudiendo afectar la compatibilidad.• Errores: Desviaciones no intencionales debido a fallos en la implementación o mala interpretación del estándar.
Uso de características no estándar	Introducir características no estándar disminuye la portabilidad y usabilidad futura del programa. Es recomendable documentar y agregar dichas características.

El éxito de un lenguaje de programación no solo depende de su capacidad para poder revolucionar, sino también de la solidez de sus estándares (La estandarización busca un equilibrio entre la innovación y estabilidad, garantizando que las herramientas desarrolladas sean confiables y así mismo duraderas.)

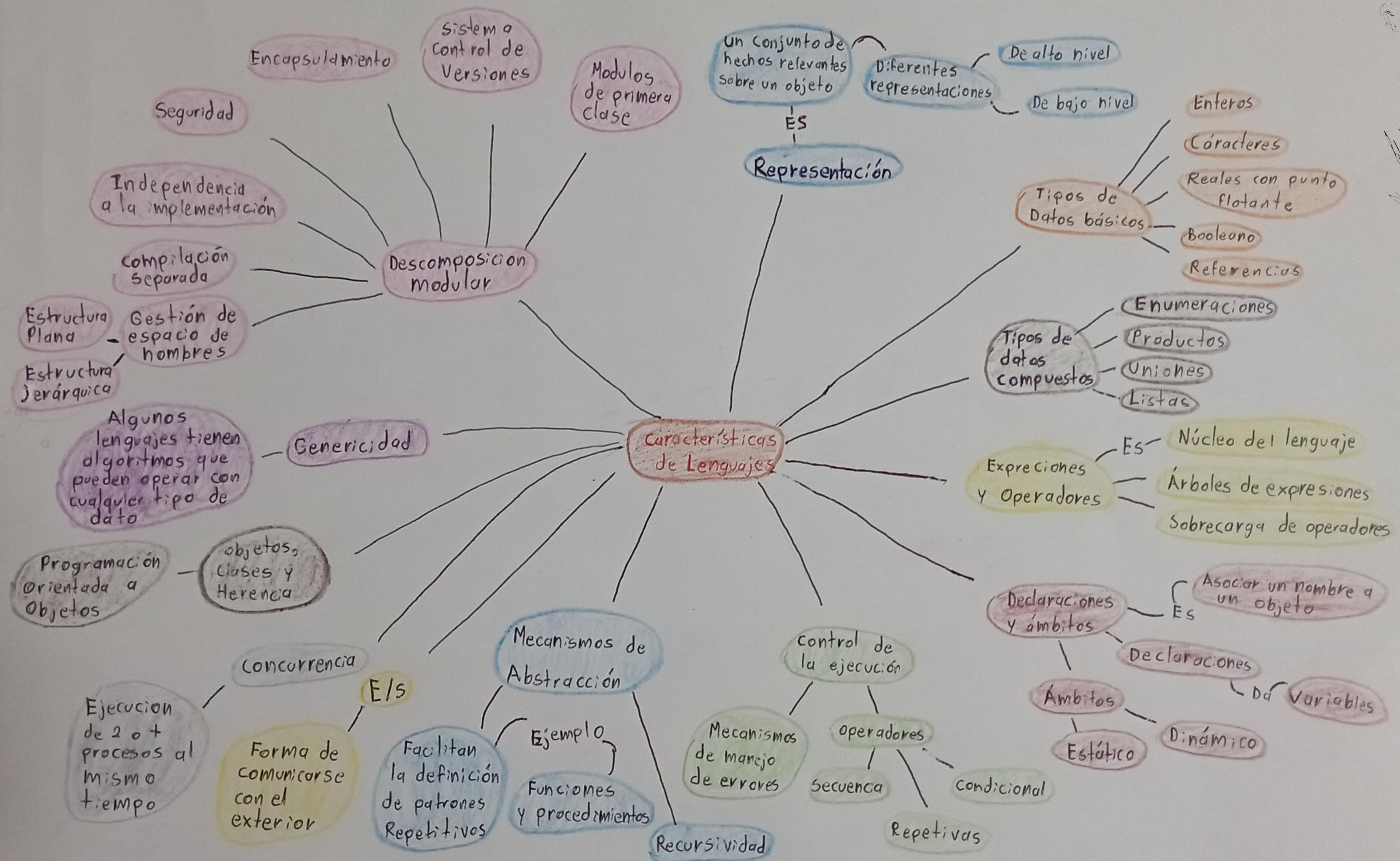


Conclusión:

• Es muy evidente que la estandarización de un lenguaje de programación es un proceso muy crucial para poder garantizar su uniformidad y portabilidad en diferentes entornos y sistemas. Un estándar formal proporciona una guía clara sobre la sintaxis y semántica del lenguaje, permitiendo que las implementaciones sean compatibles y consistentes. Pero sin embargo, el proceso de estandarización es bastante complejo por lo tanto puede llevar varios años, debido a las múltiples interpretaciones y dialectos que deben unificarse.

También hay que recalcar que aunque las extensiones y modificaciones pueden enriquecer el lenguaje, también llevan riesgos para la compatibilidad y la portabilidad de los programas. Los programadores deben ser cautelosos al incorporar características no estándar ya que esto puede limitar la usabilidad futura del software.

En resumen se puede decir que: Al adherirse al estándar es esencial para poder maximizar la interoperabilidad y la longevidad de las aplicaciones desarrolladas.



2.6 Familias de Lenguajes

Familia	Principales Características	Ejemplos
Lenguajes Imperativos	<ul style="list-style-type: none"> Son inspirados en la máquina de Von Neumann y se basan en las máquinas de Turing. Enfatizan la secuencia de instrucciones que modifica el estado del programador. Hacen uso de sentencias de control como bucles, además permiten el uso de subprogramas. Lenguajes como FORTRAN no soportaban recursividad en versiones tempranas. 	<p>FORTRAN: Primer lenguaje enfocado en la descripción de fórmulas matemáticas y usado en entornos científicos.</p> <p>COBOL: Orientado en gestión empresarial y con sintaxis inspirada en lenguaje natural.</p> <p>BASIC: Creado con el objetivo de ser utilizable por cualquiera aun si no es informático, ahora es Visual Basic.</p> <p>PASCAL: Simple y eficiente, desarrollado para la educación.</p>
Lenguajes Funcionales	<ul style="list-style-type: none"> Se basan en la noción de función matemática. Permiten el definir funciones de orden superior (que reciben y devuelven otras funciones). Hacen uso de características como la evaluación ansiosa o perezosa. 	<p>LISP: Es usado en IA y permite metaprogramación, además es muy expresivo.</p> <p>Scheme: Es un dialecto de LISP simplificado y se usa en educación.</p> <p>Haskell: Un lenguaje puramente funcional y con evaluación perezosa.</p> <p>ML: Lenguaje de propósito general, contiene chequeo estático de tipos y polimorfismo.</p>

Lenguajes Orientados a Objetos

- Se basan en objetos que encapsulan datos y comportamientos.
- Facilitan el reutilizar código mediante herencias.
- Soportan clases, subclases, herencias, polimorfismo y encapsulamiento.
- Algunas lenguajes de esta familia han evolucionado para incluir manejo de memoria dinámico y concurrente.

Simula: Desarrollado para simulación de eventos discretos y paralelo en el uso de clases y subclases.
Smalltalk: El primer lenguaje puramente orientado a objetos.

C++: Creado como extensión de C con orientación a objetos, con alta eficiencia y soporte para generidad.

Java: Con sintaxis inspirada en C++ y popularizado por la portabilidad gracias a la JVM y su uso en aplicaciones web.

Lenguajes de Programación Logica

- Basados en relaciones lógicas y reglas para búsqueda de soluciones.
- Hacen uso de algoritmos de resolución y backtracking para encontrar soluciones.
- Destacan en IA y procesamiento de lenguaje natural.
- Son capaces de unificar la programación funcional y lógica.

Prolog: Se usa en aplicaciones de inteligencia artificial y en el procesamiento de lenguaje natural.

Curry: Esta diseñado para combinar la programación lógica y funcional, hace uso de la técnica narrowing para resolver problemas.

2.7 Lenguajes de Dominio Específico (DSL)

- Definición
 - Es un lenguaje de programación o especificación ejecutable.
 - Se enfocan en resolver problemas en un dominio concreto.
 - Es eficiente en su ámbito debido a la potencia expresiva específica que ofrece.

Comparación

Lenguajes de Dominio Específico (DSL)

- Brindan soluciones optimizadas para problemas específicos.
- Suelen ser pequeñas y declarativos.
- Se pueden empollar en lenguajes generales.
- Se enfocan en programación por usuarios finales.

Lenguajes de Propósito General (GPL)

- Brindan soluciones generales, pero no siempre óptimas.
- Ofrecen mayor versatilidad.

Tamaño y complejidad: Son pequeños y con notaciones limitadas, además pueden incluir un sublenguaje de propósito general.

Declaratividad: Suelen ser declarativos y de especificación.

Generación de Aplicaciones: Algunos incluyen compiladores para la creación de aplicaciones.

Programación por Usuarios Finales: Se diseñan para ser usados por personas sin gran conocimiento en programación.

2.7 Lenguajes de Dominio Específico (DSL)

Técnicas de Desarrollo

Lenguaje Independiente

- Creación y procesamiento como de un lenguaje ordinario.
- Este requiere de un diseño completo e implementación de intérprete / compilador.

Empaquetado en Lenguaje General

- Lenguajes como Haskell permiten su integración.
- Se aprovechar de la versatilidad del lenguaje anfitrión.

Preprocesamiento o Macros

- Hace uso de preprocesadores para la transformación del lenguaje específico en el anfitrión.
- Un ejemplo de esto es el preprocesador de C++.

Intérprete extensible

- Permite las modificaciones del intérprete para analizar diferentes lenguajes.

Referencias

"INTÉRPRETES Y DISEÑO DE LENGUAJES DE PROGRAMACIÓN"

José Emilio Labra Gayo Juan

Manuel Cueva Lovelle

Raúl Izquierdo Castanedo

Aquilino Adolfo Juan Fuente

Ma Cándida Luengo Díez Francisco

Ortín Soler

Profesores de la Universidad de Oviedo Apartado de correos 1910, 33080 - Oviedo (España).

Teléfono

+34609108131.

Editorial: SERVITEC ISBN: 84-688-4210-9

Links de videos

Sección 2.4

https://youtu.be/t0cYGJWqhiY?si=_pc2Qic229Qg9659

Sección 2.8

<https://youtu.be/3grVBbqqhZE>