Contenido

[PRACTICA 1 5](#_Toc43379222)

[1) Evolución de los lenguajes de programación en la historia 5](#_Toc43379223)

[3) ¿Qué atributos debería tener un buen lenguaje de programación? Por ejemplo, ortogonalidad, expresividad, legibilidad, simplicidad, etc. De al menos un ejemplo de un lenguaje que cumple con las características citadas. 5](#_Toc43379224)

[5) Describa de ADA: | Tipos de datos | Tipos abstractos de datos – paquetes |Estructuras de datos | Manejo de excepciones | Manejo de concurrencia 5](#_Toc43379225)

[6) ¿Para qué fue creado Java? ¿Qué cambios le introdujo a la Web? ¿Java es un lenguaje dependiente de la plataforma en dónde se ejecuta? ¿Porqué? 5](#_Toc43379226)

[7) ¿Sobre que lenguajes está basado Java? 5](#_Toc43379227)

[8) En Java, ¿Qué son los applets? ¿Qué son los servlets? 5](#_Toc43379228)

[9) ¿Cómo es la estructura de un programa escrito en C? ¿Existe anidamiento de funciones? (spoiler: no existe) 5](#_Toc43379229)

[10) Describa el manejo de expresiones de C 5](#_Toc43379230)

[11) Sobre Python - RUBY – PHP, ¿Qué tipo de programas se pueden escribir con cada uno de estos lenguajes? ¿A que paradigma responde cada uno? ¿Qué características determinan la pertenencia a cada paradigma? 5](#_Toc43379231)

[12) Cite otras características importantes de Python, Ruby, PHP, Gobstone y Processing. Por ejemplo: tipado de datos, como se organizan los programas, etc. 5](#_Toc43379232)

[13) Sobre Javascript, ¿A qué tipo de paradigma corresponde este lenguajes? ¿A qué tipo de Lenguaje pertenece? 5](#_Toc43379233)

[14) Cite otras características importantes de javascript. Tipado de datos, excepciones, variables, etc 5](#_Toc43379234)

[PRACTICA 2 (SINTÁXIS Y GRAMÁTICA) 5](#_Toc43379235)

[2) ¿Cuál es la importancia de la sintaxis para un lenguaje? ¿Cuáles son sus elementos? 5](#_Toc43379236)

[3) Explique a qué se denomina regla lexicográfica y regla sintáctica 5](#_Toc43379237)

[4) ¿En la definición de un lenguaje, a qué se llama palabra reservadas? ¿A qué son equivalentes en la definición de una gramática? 5](#_Toc43379238)

[5)b) Indique porqué es ambigua la gramática del ejemplo 5](#_Toc43379239)

[PRACTICA 3 (SEMÁNTICA) 5](#_Toc43379240)

[1) ¿Qué define la semántica? 5](#_Toc43379241)

[2) a. ¿Qué significa compilar un programa? b. Describa brevemente cada uno de los pasos necesarios para compilar un programa. c. ¿En qué paso interviene la semántica y cual es su importancia dentro de la compilación? 5](#_Toc43379242)

[3) Mostrar las diferencias básicas, ventajas y desventajas de un lenguaje compilado vs uno interpretado. 6](#_Toc43379243)

[4) Explique claramente la diferencia entre un error sintáctico y uno semántico. Ejemplifique cada caso. 6](#_Toc43379244)

[6) Explique cuál es la semántica para las variables predefinidas en lenguaje Ruby self y nil. ¿Qué valor toman; cómo son usadas por el lenguaje? 6](#_Toc43379245)

[7) Determine la semántica de null y undefined para valores en javascript.¿Qué diferencia hay entre ellos? 6](#_Toc43379246)

[8) Determine la semántica de la sentencia break en C, PHP, javascript y Ruby. Cíte las características más importantes de esta sentencia para cada lenguaje 6](#_Toc43379247)

[9) Defina el concepto de ligadura y su importancia respecto de la semántica de un programa. ¿Qué diferencias hay entre ligadura estática y dinámica? Cite ejemplos 6](#_Toc43379248)

[PRACTICA 4 (VARIABLES) 6](#_Toc43379249)

[2) a. Indique cuales son las diferentes formas de inicializar una variable en el momento de la declaración de la misma. b. Analice en los lenguajes: Java, C, Phyton y Ruby las diferentes formas de inicialización de variables que poseen. Realice un cuadro comparativo de esta característica. 6](#_Toc43379250)

[3) Explique a.Variable estática. b. Variable automática o semiestática. c. Variable dinámica. d. Variable semidinámica. Investigue sobre que tipos de variables respecto de su l-valor hay en los lenguajes C y Ada. 6](#_Toc43379251)

[4) a. ¿A qué se denomina variable local y a qué se denomina variable global? b. ¿Una variable local puede ser estática respecto de su l-valor? c. Una variable global ¿siempre es estática? d. Indique que diferencia hay entre una variable estática respecto de su l-valor y una constante 6](#_Toc43379252)

[5) a. Explique variables numericas y comunes 6](#_Toc43379253)

[11) que es el tipo de dato de una variable 6](#_Toc43379254)

[13) ¿que puede condicionar el nombre de una variable? a) Su tiempo de vida. b) Su alcance. c) Su r-valor. d) Su tipo. Justifique 6](#_Toc43379255)

[PRACTICA 5 (PILAS DE EJECUCIÓN - UNIDADES) 6](#_Toc43379256)

[1) Explique claramente cual es la utilidad del registro de activación y que representan cada una de sus partes 6](#_Toc43379257)

[PRACTICA 6 (PILAS DE EJECUCIÓN CON PARÁMETROS) 6](#_Toc43379258)

[1) Explique brevemente los siguientes conceptos ● Parámetro ● Parámetro real ● Parámetro formal ● Ligadura posicional ● Ligadura por palabra clave o nombre 6](#_Toc43379259)

[2) Clasifica los tipos de parámetros en IN, OUT e IN/OUT 6](#_Toc43379260)

[3) a. Tipo de pasaje de parámetros según el lenguaje (para ADA, C, Rubi, Java y Python) b. Explique por qué Ada es más seguro que Pascal, respecto al pasaje de parámetros en las funciones. c. Explique cómo maneja Ada los tipos de parámetros in-out de acuerdo al tipo de dato. 7](#_Toc43379261)

[10) a. Explique cómo simularía en Pascal el pasaje por valor-resultado 7](#_Toc43379262)

[PRACTICA 7 (TIPOS DE DATOS) 7](#_Toc43379263)

[1) 1. ¿Qué es un sistema de tipos y cuál es su principal función? 2. Definir y contrastar las definiciones de un sistema de tipos fuerte y débil. 3. Caracterizar el tipado como estático o dinámico 7](#_Toc43379264)

[2) 1. Dar una definición de tipo de dato. 2. ¿Qué es un tipo predefinido elemental? 3. ¿Qué es un tipo definido por el usuario? 7](#_Toc43379265)

[3) 1. Dar una breve definición de: producto cartesiano (o product type), correspondencia finita, uniones (o sum type) y tipos recursivos. 2. En los ejemplos que se muestran decir a que tipos de dato corresponde cada ejemplo. 7](#_Toc43379266)

[4) 1. Definir mutabilidad e inmutabilidad respecto a un dato (en Python y Ruby) y explicar que es la operación #freeze en los objetos de Ruby. 7](#_Toc43379267)

[5) 1. ¿Permite C tomar el l-valor de las variables? 2. ¿Qué problemas existen en el manejo de punteros? 7](#_Toc43379268)

[6) 1. ¿Qué características debe cumplir una unidad para que sea un TAD? 2. Dar algunos ejemplos de TAD en lenguajes como ADA, Java, Python, entre otros. 7](#_Toc43379269)

[PRACTICA 8 (EXCEPCIONES) 7](#_Toc43379270)

[1) Explique claramente a qué se denomina excepción 7](#_Toc43379271)

[2) ¿Qué debería proveer un lenguaje para el manejo de las excepciones? ¿Todos los lenguajes lo proveen? 7](#_Toc43379272)

[3) ¿Qué ocurre cuando un lenguaje no provee manejo de excepciones? ¿Se podría simular? Explique cómo lo haría 7](#_Toc43379273)

[4) Explique los diferentes modelos que existen para definir que hacer luego de manejar una excepción. Da ejemplos de lenguajes y diga que modelo es más inseguro y por qué. 7](#_Toc43379274)

[5) Comparación entre CLU, PL/1, ADA en cuanto a la propagación de los errores 7](#_Toc43379275)

[9) Indique diferencias y similitudes entre Phyton y Java con respecto al manejo de excepciones. 7](#_Toc43379276)

[10) Modelo de excepciones de Ruby y sus instrucciones 7](#_Toc43379277)

[11) Modelo de excepciones de Javascript y sus instrucciones 7](#_Toc43379278)

[PRACTICA 9 7](#_Toc43379279)

[1) Diferencia entre sentencia simple y compuesta 7](#_Toc43379280)

[2) Asignación en C 8](#_Toc43379281)

[3) ¿Una expresión de asignación puede producir efectos laterales que afecten al resultado final, dependiendo de cómo se evalúe? 8](#_Toc43379282)

[4) Defina circuito corto y circuito largo 8](#_Toc43379283)

[5) ¿Qué regla define Delphi, Ada y C para la asociación del else con el if correspondiente? ¿Cómo lo maneja Python? 8](#_Toc43379284)

[6) ¿Cuál es la construcción para expresar múltiples selección que implementa C? ¿Trabaja de la misma manera que la de Pascal, ADA o Python? 8](#_Toc43379285)

[7) ¿Qué pasa si modificas el índice de un for en ADA y Pascal? 8](#_Toc43379286)

[9) ¿Qué diferencia existe entre el generador YIELD de Python y el return de una función? 8](#_Toc43379287)

[10) Describa la instrucción map en javascript y sus alternativas. 8](#_Toc43379288)

[11) Espacio de nombres en Python y PHP. 8](#_Toc43379289)

[PRACTICA 10 8](#_Toc43379290)

[1) ¿Qué es un programa escrito en un lenguaje funcional? ¿Qué rol cumple la computadora? 8](#_Toc43379291)

[2) ¿Cómo se define el lugar donde se definen las funciones en un lenguaje funcional? 8](#_Toc43379292)

[3) ¿Cuál es el concepto de variables en los lenguajes funcionales? 8](#_Toc43379293)

[4) ¿Qué es una expresión en un lenguaje funcional? ¿Su valor de qué depende? 8](#_Toc43379294)

[5) ¿Cuál es la forma de evaluación que utilizan los lenguajes funcionales? 8](#_Toc43379295)

[6) ¿Un lenguaje funcional es fuertemente tipado? ¿Qué tipos existen? ¿Por qué? 8](#_Toc43379296)

[7) ¿Cómo definiría un programa escrito en POO? 8](#_Toc43379297)

[8) Elementos más imporantes de POO 8](#_Toc43379298)

[9) Cuál es el segundo nivel de abstracción de la POO 8](#_Toc43379299)

[10) ¿Qué tipos de herencias hay? Cuál usa Smalltalk y C++ 8](#_Toc43379300)

[11) En el paradigma lógico ¿Qué representa una variable? ¿y las constantes? 8](#_Toc43379301)

[12) ¿Cómo se escribe un programa en un lenguaje lógico? 8](#_Toc43379302)

[14) Describa las características más importantes de los Lenguajes Basados en Scripts. Mencione diferentes lenguajes que utilizan este concepto. ¿En general, qué tipificación utilizan? 8](#_Toc43379303)

[15) Breve descripción de los paradigmas dirigido por eventos y orientado a aspectos 8](#_Toc43379304)

# PRACTICA 1

## 1) Evolución de los lenguajes de programación en la historia

## 3) ¿Qué atributos debería tener un buen lenguaje de programación? Por ejemplo, ortogonalidad, expresividad, legibilidad, simplicidad, etc. De al menos un ejemplo de un lenguaje que cumple con las características citadas.

## 5) Describa de ADA: | Tipos de datos | Tipos abstractos de datos – paquetes |Estructuras de datos | Manejo de excepciones | Manejo de concurrencia

## 6) ¿Para qué fue creado Java? ¿Qué cambios le introdujo a la Web? ¿Java es un lenguaje dependiente de la plataforma en dónde se ejecuta? ¿Porqué?

## 7) ¿Sobre que lenguajes está basado Java?

## 8) En Java, ¿Qué son los applets? ¿Qué son los servlets?

## 9) ¿Cómo es la estructura de un programa escrito en C? ¿Existe anidamiento de funciones? (spoiler: no existe)

## 10) Describa el manejo de expresiones de C

## 11) Sobre Python - RUBY – PHP, ¿Qué tipo de programas se pueden escribir con cada uno de estos lenguajes? ¿A que paradigma responde cada uno? ¿Qué características determinan la pertenencia a cada paradigma?

## 12) Cite otras características importantes de Python, Ruby, PHP, Gobstone y Processing. Por ejemplo: tipado de datos, como se organizan los programas, etc.

## 13) Sobre Javascript, ¿A qué tipo de paradigma corresponde este lenguajes? ¿A qué tipo de Lenguaje pertenece?

## 14) Cite otras características importantes de javascript. Tipado de datos, excepciones, variables, etc

# PRACTICA 2 (SINTÁXIS Y GRAMÁTICA)

## 2) ¿Cuál es la importancia de la sintaxis para un lenguaje? ¿Cuáles son sus elementos?

## 3) Explique a qué se denomina regla lexicográfica y regla sintáctica

## 4) ¿En la definición de un lenguaje, a qué se llama palabra reservadas? ¿A qué son equivalentes en la definición de una gramática?

## 5)b) Indique porqué es ambigua la gramática del ejemplo

# PRACTICA 3 (SEMÁNTICA)

## 1) ¿Qué define la semántica?

## 2) a. ¿Qué significa compilar un programa? b. Describa brevemente cada uno de los pasos necesarios para compilar un programa. c. ¿En qué paso interviene la semántica y cual es su importancia dentro de la compilación?

## 3) Mostrar las diferencias básicas, ventajas y desventajas de un lenguaje compilado vs uno interpretado.

## 4) Explique claramente la diferencia entre un error sintáctico y uno semántico. Ejemplifique cada caso.

## 6) Explique cuál es la semántica para las variables predefinidas en lenguaje Ruby self y nil. ¿Qué valor toman; cómo son usadas por el lenguaje?

## 7) Determine la semántica de null y undefined para valores en javascript.¿Qué diferencia hay entre ellos?

## 8) Determine la semántica de la sentencia break en C, PHP, javascript y Ruby. Cíte las características más importantes de esta sentencia para cada lenguaje

## 9) Defina el concepto de ligadura y su importancia respecto de la semántica de un programa. ¿Qué diferencias hay entre ligadura estática y dinámica? Cite ejemplos

# PRACTICA 4 (VARIABLES)

## 2) a. Indique cuales son las diferentes formas de inicializar una variable en el momento de la declaración de la misma. b. Analice en los lenguajes: Java, C, Phyton y Ruby las diferentes formas de inicialización de variables que poseen. Realice un cuadro comparativo de esta característica.

## 3) Explique a.Variable estática. b. Variable automática o semiestática. c. Variable dinámica. d. Variable semidinámica. Investigue sobre que tipos de variables respecto de su l-valor hay en los lenguajes C y Ada.

## 4) a. ¿A qué se denomina variable local y a qué se denomina variable global? b. ¿Una variable local puede ser estática respecto de su l-valor? c. Una variable global ¿siempre es estática? d. Indique que diferencia hay entre una variable estática respecto de su l-valor y una constante

## 5) a. Explique variables numericas y comunes

## 11) que es el tipo de dato de una variable

## 13) ¿que puede condicionar el nombre de una variable? a) Su tiempo de vida. b) Su alcance. c) Su r-valor. d) Su tipo. Justifique

# PRACTICA 5 (PILAS DE EJECUCIÓN - UNIDADES)

## 1) Explique claramente cual es la utilidad del registro de activación y que representan cada una de sus partes

# PRACTICA 6 (PILAS DE EJECUCIÓN CON PARÁMETROS)

## 1) Explique brevemente los siguientes conceptos ● Parámetro ● Parámetro real ● Parámetro formal ● Ligadura posicional ● Ligadura por palabra clave o nombre

## 2) Clasifica los tipos de parámetros en IN, OUT e IN/OUT

## 3) a. Tipo de pasaje de parámetros según el lenguaje (para ADA, C, Rubi, Java y Python) b. Explique por qué Ada es más seguro que Pascal, respecto al pasaje de parámetros en las funciones. c. Explique cómo maneja Ada los tipos de parámetros in-out de acuerdo al tipo de dato.

## 10) a. Explique cómo simularía en Pascal el pasaje por valor-resultado

# PRACTICA 7 (TIPOS DE DATOS)

## 1) 1. ¿Qué es un sistema de tipos y cuál es su principal función? 2. Definir y contrastar las definiciones de un sistema de tipos fuerte y débil. 3. Caracterizar el tipado como estático o dinámico

## 2) 1. Dar una definición de tipo de dato. 2. ¿Qué es un tipo predefinido elemental? 3. ¿Qué es un tipo definido por el usuario?

## 3) 1. Dar una breve definición de: producto cartesiano (o product type), correspondencia finita, uniones (o sum type) y tipos recursivos. 2. En los ejemplos que se muestran decir a que tipos de dato corresponde cada ejemplo.

## 4) 1. Definir mutabilidad e inmutabilidad respecto a un dato (en Python y Ruby) y explicar que es la operación #freeze en los objetos de Ruby.

## 5) 1. ¿Permite C tomar el l-valor de las variables? 2. ¿Qué problemas existen en el manejo de punteros?

## 6) 1. ¿Qué características debe cumplir una unidad para que sea un TAD? 2. Dar algunos ejemplos de TAD en lenguajes como ADA, Java, Python, entre otros.

# PRACTICA 8 (EXCEPCIONES)

## 1) Explique claramente a qué se denomina excepción

## 2) ¿Qué debería proveer un lenguaje para el manejo de las excepciones? ¿Todos los lenguajes lo proveen?

## 3) ¿Qué ocurre cuando un lenguaje no provee manejo de excepciones? ¿Se podría simular? Explique cómo lo haría

## 4) Explique los diferentes modelos que existen para definir que hacer luego de manejar una excepción. Da ejemplos de lenguajes y diga que modelo es más inseguro y por qué.

## 5) Comparación entre CLU, PL/1, ADA en cuanto a la propagación de los errores

## 9) Indique diferencias y similitudes entre Phyton y Java con respecto al manejo de excepciones.

## 10) Modelo de excepciones de Ruby y sus instrucciones

## 11) Modelo de excepciones de Javascript y sus instrucciones

# PRACTICA 9

## 1) Diferencia entre sentencia simple y compuesta

## 2) Asignación en C

## 3) ¿Una expresión de asignación puede producir efectos laterales que afecten al resultado final, dependiendo de cómo se evalúe?

## 4) Defina circuito corto y circuito largo

## 5) ¿Qué regla define Delphi, Ada y C para la asociación del else con el if correspondiente? ¿Cómo lo maneja Python?

## 6) ¿Cuál es la construcción para expresar múltiples selección que implementa C? ¿Trabaja de la misma manera que la de Pascal, ADA o Python?

## 7) ¿Qué pasa si modificas el índice de un for en ADA y Pascal?

## 9) ¿Qué diferencia existe entre el generador YIELD de Python y el return de una función?

## 10) Describa la instrucción map en javascript y sus alternativas.

## 11) Espacio de nombres en Python y PHP.

# PRACTICA 10

## 1) ¿Qué es un programa escrito en un lenguaje funcional? ¿Qué rol cumple la computadora?

## 2) ¿Cómo se define el lugar donde se definen las funciones en un lenguaje funcional?

## 3) ¿Cuál es el concepto de variables en los lenguajes funcionales?

## 4) ¿Qué es una expresión en un lenguaje funcional? ¿Su valor de qué depende?

## 5) ¿Cuál es la forma de evaluación que utilizan los lenguajes funcionales?

## 6) ¿Un lenguaje funcional es fuertemente tipado? ¿Qué tipos existen? ¿Por qué?

## 7) ¿Cómo definiría un programa escrito en POO?

## 8) Elementos más imporantes de POO

## 9) Cuál es el segundo nivel de abstracción de la POO

## 10) ¿Qué tipos de herencias hay? Cuál usa Smalltalk y C++

## 11) En el paradigma lógico ¿Qué representa una variable? ¿y las constantes?

## 12) ¿Cómo se escribe un programa en un lenguaje lógico?

## 14) Describa las características más importantes de los Lenguajes Basados en Scripts. Mencione diferentes lenguajes que utilizan este concepto. ¿En general, qué tipificación utilizan?

## 15) Breve descripción de los paradigmas dirigido por eventos y orientado a aspectos