La programación convencional priorizaba Sus principales problemáticas incluyen: el desarrollo rápido sin considerar el dificultad en la depuración, modificación mantenimiento. Esto generaba programas y ampliación del código, baja sin estructura definida, con instrucciones transportabilidad, falta de documentación secuenciales difíciles de entender y y redundancia de código, lo que complica modificar por alguien hizo el programa. su mantenimiento y comprensión. Divide un programa en módulos o Es un paradigma aplicable a subprogramas, cada uno con una lenguajes de bajo y alto nivel. En tarea específica y bien definida. Estos lenguajes de alto nivel, se conoce La técnica Bottom-Up, o se agrupan por funcionalidad y se programación ascendente, se Enfoque como programación funcional. Se desarrollan de forma independiente, basa en reducir la repetición de diferencia del enfoque descendente nonolítico facilitando el análisis y la codificación. expresiones agrupándolas en (Top-Down) en la forma de funciones o procedimientos enumerar y desarrollar los módulos. reutilizables. Técnicas básicas para el En Bottom-Up, se programan desarrollo de sistemas de Método primero los módulos inferiores y se **Programación** programas de computadoras ascendente rogramación avanza hacia el superior. Aunque la procedimental modular estructura del diseño es la misma en ambos enfoques, la diferencia radica en el orden de implementación del código. Optimiza el tamaño del código sin afectar significativamente la velocidad de ejecución, siempre Otras técnica Método que los datos se almacenen en Comienza con la definición del módulo descendente de diseño memoria. Pero lograr una principal, seguido de la creación de implementación completamente módulos específicos organizados pura es difícil. jerárquicamente. Programación Este enfoque suele representarse POO estructurada mediante un diagrama de diseño **Enfoque** estructurado, que indica el flujo de control y la relación entre los módulos. El programa se organiza en objetos, entidades autosuficientes con características relacionadas y Desarrollada en las décadas de 1960 y diseñadas para realizar tareas 1970, fue impulsada por Niklaus Wirth específicas. Cada uno tiene una con Pascal y formalizada por E. W. funcionalidad que es accesible por sus Dijkstra. Se basa en la programación usuarios, pero su implementación Se compone de 3 modular, promoviendo un código más interna está oculta. bloques principales: organizado y fácil de mantener. Entrada, proceso y salida. Entrada de datos: El programador Salida de resultados: El Algoritmo de resolución: El define las fuentes y formatos de programador determina qué programador diseña el algoritmo los datos necesarios para el salidas se necesitan, su que procesa los datos y genera los formato (pantalla, impresora, programa, asegurando que todas resultados esperados. Esto incluye las entradas estén disponibles y disco) y la frecuencia con la la creación de un modelo para la sean procesadas correctamente. que se deben generar, como resolución del problema y la listas, tablas, informes o expresión del algoritmo en gráficos. pseudocódigo, diagramas de flujo o diagramas estructurados.

LCC- Autómatas y compiladores - 6°3 - Roberto Estrada Tovar

Bibliografía:

En Fundamentos generales de programación, de Luis Joyanes Aguilar, 33-37. México: Mc Graw-Hill, 2013. Aquilar, Luis joyanes. En Metodologia de la programación Diagramas de flujo, Algoritmos y programación Estructurada, 66-88. Mexico: Mc Graw-Hill, 1987. En Fundamentos de Programación de Manuel Santos. Ismael Patiño. Raul Carrasco, 37-40. México: AlfaOmega-Rama, 2006. Chorda, Gloria de Antonio. Ramon. «Metodologia de la Programación.» 39-41. España: Ra-ma, s.f. En Lenguaje Ensamblador para Mircrocomputadoras IBM, de J. Terry Godfrey, 73-76. México: Prentice Hall, 1991. En Introducción a la computación y a la programación Estructurada, de Guillermo Levine, 115-118, Mexico, 1989. En Sistema Operativos y

Compiladores, de Jesus Salas

Graw-Hill, 1992.

Parilla, 149,150,173,174. México : Mc