

Actividad | 1# | Diagramas de Paradigma

Estructural

Análisis y Diseño de Sistemas

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo García

ALUMNO: Casandra Montserrat Ortiz Cortes G-1

FECHA: 26/02/2024

Índice

Introducción.....1

Descripción.....2

Justificación.....3

Desarrollo.....4

- Diagramas de contexto
- Diagramas de flujo de datos

Conclusión.....5

Referencia

INTRODUCCIÓN

La programación estructurada es una disposición en el diseño y construcción de programas informáticos, con el objetivo de mejorar la gestión de su complejidad, teniendo en cuenta las peculiaridades del pensamiento humano. Todo programa se basa en un algoritmo y puede representarse mediante un esquema lógico. Surgió por la creciente complejidad de los programas computarizados. Se ha convertido en un nuevo esquema de programación que creó nuevas técnicas para poder producir programas seguros de operar por más tiempo. Gracias al vertiginoso desarrollo del procesamiento de datos computarizado, a fines de los años 60 era posible resolver problemas cada vez más complejos. Sin embargo, comprender, depurar y modificar estos programas presentaba dificultades que los hacían poco confiables.

Los programas de computación grandes realizados para estos problemas tenían códigos fuentes tan largos y desestructurados que se habían vuelto bastante complicados de acceder y navegar con cierta lógica, incluso para sus propios autores.

DESCRIPCIÓN

Dijkstra ideó una teoría vinculada a la programación estructurada, la cual indicaba que al diseñar cualquier programa es conveniente tomar en cuenta los siguientes fundamentos:

– El teorema estructural, que expresa que se puede compilar cualquier programa utilizando solo tres estructuras de control esenciales: estructura secuencial, estructura de alternativas y estructura repetitiva.

– Al delinear los programas se exhorta a aplicar la técnica descendente, llamada también de arriba hacia abajo.

La programación estructurada se basa en módulos funcionales bien definidos, jerarquizados según la naturaleza específica del problema. Esta programación es un método independiente del lenguaje de programación, actuando más bien como un estilo de programar.

Es una forma de idear los programas siguiendo reglas bien establecidas, utilizando para ello un cierto conjunto de estructuras de control.

La programación estructurada permite que los programas se escriban en pseudocódigo, independientemente del lenguaje de la máquina, cercano al natural, convertible en cualquier lenguaje de programación.

JUSTIFICACIÓN

Casi cualquier fragmento de código de muestra en cualquier lenguaje de programación moderno será un ejemplo de programación estructurada.

Esta programación utiliza bucles, condicionales y subrutinas para controlar el flujo y modularidad para así facilitar la lectura y poder reutilizar el código.

Secuencia

Se puede ejecutar una lista de sentencias en orden, de arriba hacia abajo. En este caso se da como ejemplo un programa en lenguaje natural para hornear pan:

– Añadir levadura.

– Agregar sal.

– Mezclar.

– Añadir agua.

– Amasar.

– Dejar que se levante la masa.

– Hornear.

Repetición

Se repite un bloque de sentencias mientras una condición sea verdadera. Por ejemplo: lavar

platos

Apilar los platos en el fregadero.

Llenar el fregadero con agua jabonosa caliente.

Mientras haya más platos

Tomar plato del mostrador,

Lavar el plato,

Colocar el plato en la rejilla de drenaje.

Fin de Mientras

Limpiar el mostrador.

Enjuagar el fregadero.

Selección

Se elige como máximo una acción entre varias condiciones alternas. Por ejemplo: ordenar el correo.

Obtener el correo del buzón.

Poner todo el correo sobre la mesa.

Mientras haya correos para ordenar

Tomar un correo de la mesa.

Si el correo es personal entonces

Leerlo

de lo contrario, si el correo es una revista entonces

Ponerlo en el revistero

de lo contrario, si el correo es una factura entonces

Pagarla

de lo contrario si el correo es alguna publicidad entonces

Tirarlo en la papelera

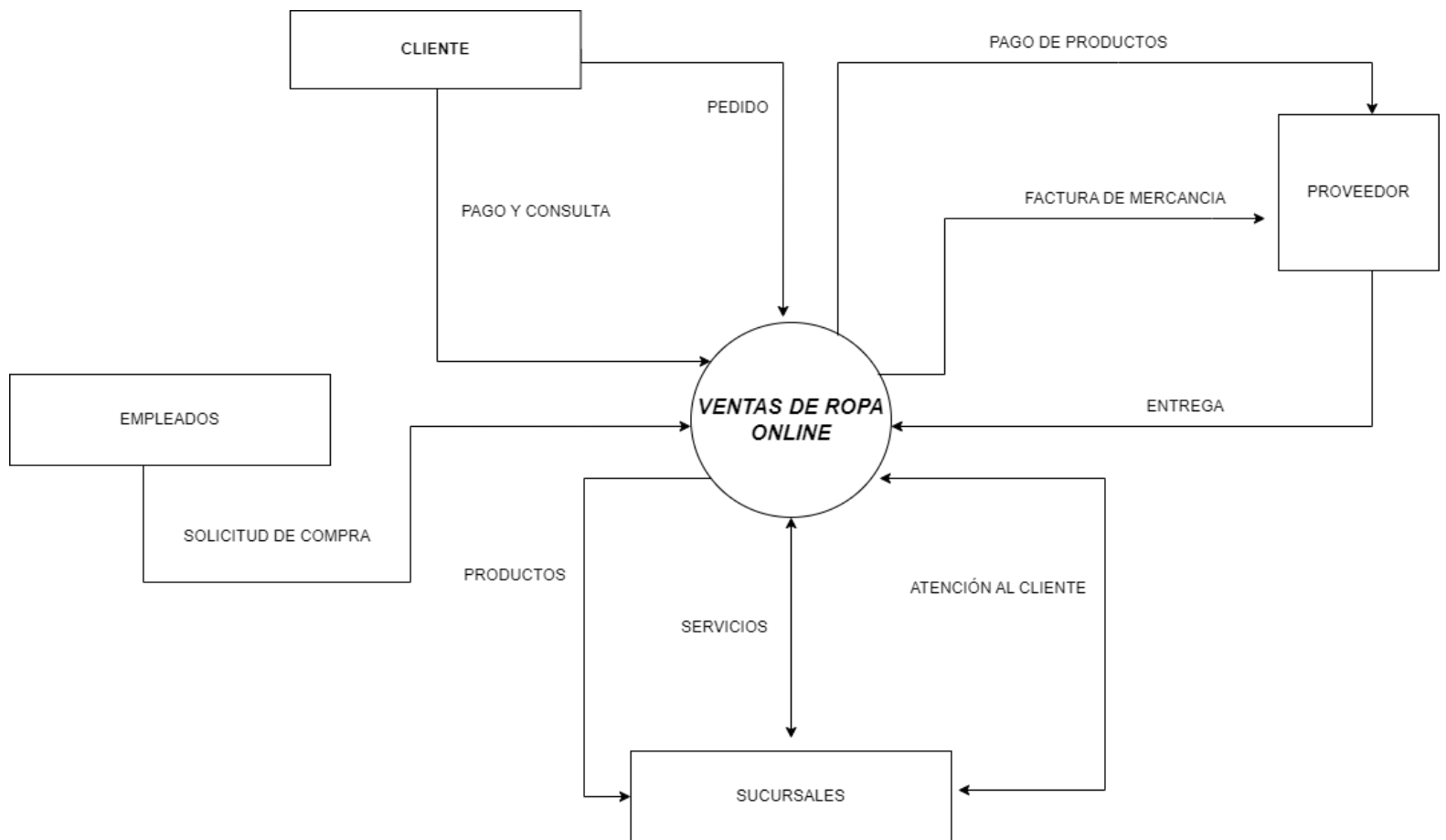
Fin de Si

Fin de Mientras.

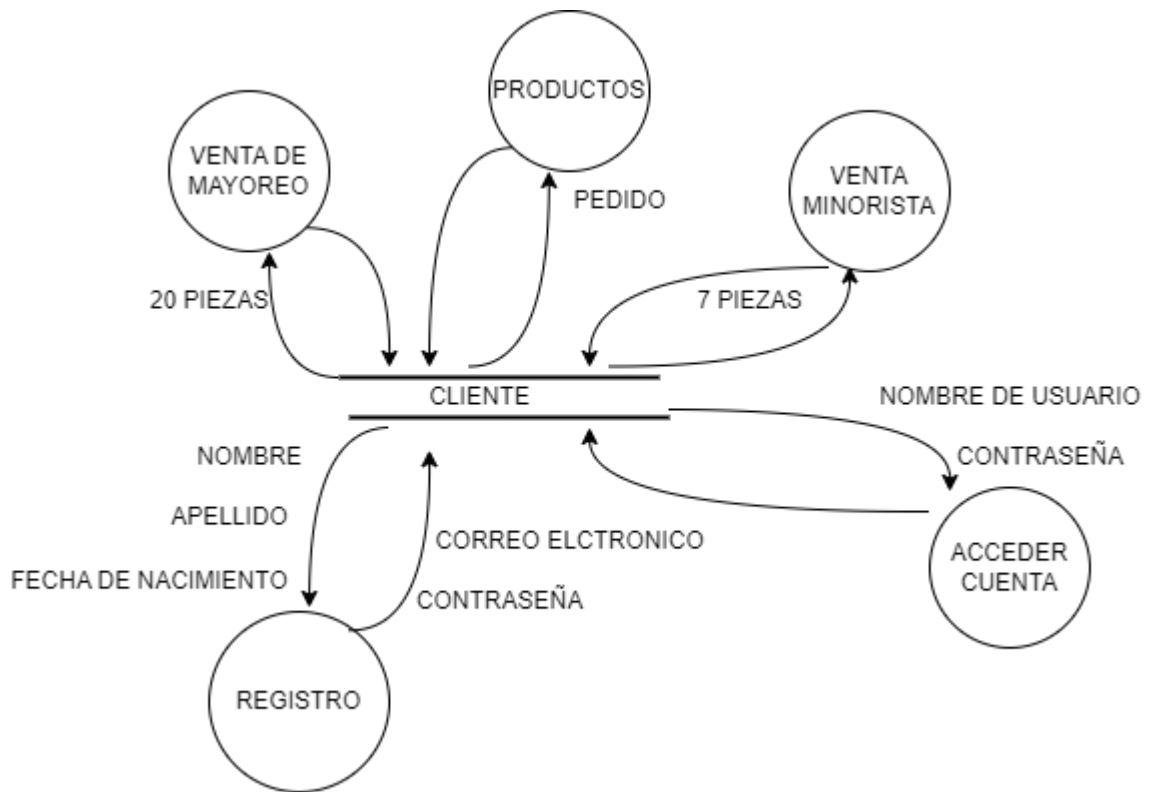
La programación estructurada reduce la posibilidad que una función afecte a otra. Esto hace que se escriban programas más claros, ya que se eliminan las variables globales para ser reemplazadas .

DESARROLLO

- Diagramas de Contexto



- Diagramas de Flujo de Datos



CONCLUSIÓN

Los paradigmas pueden influir en nuestra percepción de la realidad, nuestras decisiones y nuestras acciones en la vida cotidiana. Si tenemos paradigmas limitantes, es menos probable que busquemos nuevas opciones y posibilidades.

Los paradigmas limitantes suelen reflejarse en nuestras creencias y pensamientos automáticos.

Si notamos que nuestras creencias nos limitan y nos impiden buscar nuevas opciones y posibilidades, es probable que tengamos paradigmas limitantes.

Para ampliar nuestros paradigmas y posibilidades, es importante cuestionar nuestras creencias y exponernos a nuevas ideas y perspectivas. También podemos buscar experiencias que nos desafíen y nos permitan crecer y aprender.

Los paradigmas son modelos mentales que influyen en nuestra percepción de la realidad y nuestras decisiones en la vida cotidiana. Es importante ser conscientes de nuestros paradigmas y evaluar si nos están limitando o ampliando nuestras opciones y posibilidades. Al cuestionar y ampliar nuestros paradigmas, podemos mejorar nuestra calidad de vida y alcanzar nuestros objetivos.

Referencias

Helmut Sy Corvo. (8 de marzo de 2020). *Programación estructurada: características, ejemplos, ventajas, aplicaciones*.
Obtenido de <https://www.lifeder.com/programacion-estructurada/>