**Logotipo

Descripción generada automáticamenteUniversidad del Sur**

**Tema:**

Esquema conceptual de una base de datos

**Licenciatura en Ingeniería en Sistemas Computacionales.**

**Nombre del estudiante:**

- Edison Arquímides García Pech.

**Asesor:**

Ing. Villafaña Rejón Luis Fernando.

Fecha de entrega:

20/11/2023

**Índice**

### Punto 1 y 2 ………………………………………………………………………………. 1

### Punto 3 ……………………………………………………………………………………. 2

### Punto 4 ……………………………………………………………………………………. 3

### Conclusión ………………………………………………………………………………… 4

**Punto 1 y 2.-**

**Resumen de las características clave de los modelos el jerárquico y el orientado a objetos.**

**1. Modelo jerárquico:**

- Organiza los datos en una estructura de árbol con una única raíz y ramas que se extienden hacia abajo.

- Los datos se almacenan en registros y cada registro tiene un único padre y cero o más hijos.

- Permite una relación de uno a muchos.

- Está basado en conceptos de estructura en árbol.

- Es eficiente para aplicaciones donde la información se estructura en niveles o categorías claras.

**2. Modelo orientado a objetos:**

- Organiza los datos en objetos que contienen tanto atributos como métodos.

- Los objetos se pueden relacionar entre sí a través de mensajes o interacciones.

- Permite la encapsulación de datos y comportamiento en un único objeto.

- Permite la reutilización de código mediante la herencia y la creación de clases que actúan como plantilla para la creación de objetos.

- Es eficiente para aplicaciones donde la interacción y la representación de objetos del mundo real es importante.

Pág.1

**Punto 3.-**

**Comparar los modelos basándose en criterios como la complejidad, la eficiencia, la flexibilidad y la facilidad de uso.**

|  | **Modelo jerárquico** | **Modelo orientado a objetos** |
| --- | --- | --- |
| Complejidad | Es relativamente sencillo de entender y utilizar, ya que se basa en una estructura de árbol simple. | Puede tener una curva de aprendizaje más pronunciada debido a la necesidad de comprender los conceptos de objetos, clases, herencia, etc. |
| Eficiencia | Es eficiente para aplicaciones donde la información se estructura en niveles claros y las consultas son principalmente de tipo jerárquico. | Puede tener un mayor costo computacional debido a la necesidad de administrar objetos y sus relaciones. |
| Flexibilidad | Puede ser menos flexible en comparación con el modelo orientado a objetos, ya que no permite una relación más compleja entre los datos. | Es más flexible ya que permite crear relaciones de diferentes tipos, como la herencia y la composición, lo que facilita la reutilización y la extensibilidad del código. |
| Facilidad de uso | Es relativamente fácil de entender y usar, especialmente para aplicaciones con estructuras de datos jerárquicos simples. | Puede ser más complejo de entender y utilizar, especialmente para aquellos sin experiencia previa en programación orientada a objetos. |

Pág.2

**Punto 4.-**

**Discutir en qué caso sería más apropiado utilizar cada modelo y por qué.**

**1.- Modelo jerárquico:**

Este modelo es más apropiado cuando se necesita representar una estructura de datos en forma de árbol donde cada elemento tiene un solo padre y puede tener varios hijos. Es útil para representar relaciones de dependencia, como en un sistema de archivos, donde existe una jerarquía de directorios y archivos.

**2.- Modelo orientado a objetos:**

Este modelo es más apropiado cuando se necesita representar entidades del mundo real y sus interacciones. Permite encapsular los datos y las operaciones relacionadas en objetos que pueden tener propiedades (atributos) y comportamientos (métodos). Además, el modelo orientado a objetos facilita la reutilización de código, ya que los objetos pueden heredar características de otros objetos y pueden ser modificados y extendidos de manera flexible.

Pág.3

**Conclusión**

En conclusión, el modelo jerárquico es más adecuado cuando se necesita representar una estructura de datos en forma de árbol con una relación de uno a muchos, mientras que el modelo orientado a objetos es más apropiado cuando se necesita representar entidades del mundo real y sus interacciones, permitiendo la encapsulación de datos y comportamiento en objetos y facilitando la reutilización de código.

Pág.4