Guia #3

• Roger Calvo • Brayan Vergara • Julián González

TALLER DE PROGRAMACION

2021

**Trabajo Autónomo:**

1. Link Del Video: <https://youtu.be/grguaYNn35E>
2. **Defina el Concepto de patron de software**

Respuesta: Es una forma de estandarizar la solución de problemas comunes de diseño en el desarrollo de software algunas de sus ventajas de uso pueden ser que conforman un amplio catalogo de problemas y soluciones, simplifican el aprendizaje de las beuans practicas

1. **Explique con sus propias palabras en que consiste el principio de alta cohesión y bajo acoplamiento**

Respuesta: Alta cohesión la información que almacena una clase debe ser coherente y debe estar relacionada con la clase y existen 7 tipos de cohesión (coincidente, lógica, temporal, de procedimiento, de comunicación, de información y funcional), la idea principal de bajo acoplamiento es que las clases estén lo menos enlazadas entre si para que cuando en alguna clase se modifique en las demás tenga la mínima consecuencia para poder reutilizarse y no ser dependiente de las demás clases

1. **Enuncie y explique 5 patrones de concurrencia**

Respuesta:

* + Mutex: para la gestión de operaciones concurrentes de forma segura
  + Buffered channel: como contenedor de recursos compartidos
  + Factory Function: para inicializar los recursos compartidos
  + Flag: Para Representar el estado (abierto-cerrado) del pool
  + Thread Pool: Hilo de ejecución concurrente proporcionado por el sistema operativo

1. **¿Por qué se puede afirmar que patron Thread Pool es un patron de concurrencia?**

Respuesta: Ya que como Thread es un sistema de tiempo real se diseñan como sistemas concurrentes, y la ejecución del mismo son para tareasy manejadores de eventos, así como a su acceso compartido.

|  |  |  |
| --- | --- | --- |
| PATRON SOFTWARE | DESCRIPCION | EJEMPLO |
| Abstract Factory | Una interfaz crea conjuntos o familias de objetos relacionados sin especificar el nombre de la clase | Intenta solucionar es el crear diferente familia de objetos |
| Builder Patterns | Permite producir diferentes tipos y representaciones de un objeto utilizando el mismo código de construcción | Se usa para permitir la creación de una variedad de objetos complejos desde un objeto fuente |
| Factory Method | Proporciona una interfaz para crear objetos en una superclase, pero permite que las subclases alteren el tipo de objetos que se crearan | Cuando no conozcas de antemano las dependencias y los tipos exactos de los objetos con los que deba funcionar tu codigo |
| Prototype | Permite copiar objetos existentes sin hacer que su código dependa de sus clases | Permite que un objeto pueda crear una copia de si mismo sin conocer su clase o ningún detalle sobre como crearlos |
| Singleton | Este patrón de diseño restringe la creación de instancias de una clase a un único objeto | Se implementa creando en nuestra clase un método que crea una instancia del objeto solo si todavía no existe alguna |

**Análisis e interpretación de resultados**

¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro que hacer profesional.

* Identificamos que un Patrón de Software es “Una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución” lo cual nos sirve para implementar en la práctica en el futuro para nuestro desempeño laboral.
* Aprendimos que los patrones se clasifican en: Patrones creacionales, estructurales, de comportamiento pero que además encontramos los patrones de concurrencia los cuales ayudan a crear y mejorar una interfaz entre objetos, sincronizar la memoria compartida entre hilos, hacer que los datos sean seguros, monitorear el progreso y administrar hilos y eventos. Por tanto, nos aporta conocimiento que nos va a permitir tener la capacidad de desempeñarnos en el campo profesional y laboral en cualquier campo relacionado con nuestra carrera.
* La investigación y la lectura nos permitió conocer que Mark Richards escribió un libro llamado “Software Architecture Patterns” según el que existen 5 patrones principales de arquitectura de software: **microkernel, microservicios, arquitectura en capas, basada en eventos y basada en el espacio.** Lo cual es fundamental conocer para nuestra formación profesional y desempeño laboral en el futuro.

¿Dónde presento mayor dificultad resolviendo la guía? y ¿cómo lo resolvieron? ¿cuáles fueron las estrategias de solución?

Al realizar el trabajo para resolver la guía sólo fue un poco dispendioso seleccionar la información que encontramos al investigar y leer ya que se encuentra gran variedad de información sobre la temática, pero todos opinamos y esto nos permitió obtener conclusiones de manera adecuada y rápida puesto que nos conectamos para comunicarnos en tiempo real y así dar solución concreta a cada punto que contenía la guía. Por tanto, el aprendizaje es colaborativo y enriquecedor porque se permite a cada integrante del grupo hacer sus aportes y así aclaramos dudas.

¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro que hacer profesional.