1. Datos Generales de la asignatura

Nombre de la asignatura:	Desarrollo Back-end.
Clave de la asignatura:	DSB-2303
SATCA[1]:	1-4-5
Carrera:	Ingeniería Informática.

2. Presentación

Caracterización de la asignatura

Un patrón de diseño es una forma reutilizable de resolver un problema común. Por ello la presente asignatura permite adquirir las competencias para identificarlos y usarlos correctamente según el contexto donde se ha de desarrollar el proyecto completo o una parte de ella.

La utilización de patrones de diseño demuestra la madurez de un programador de software pues utiliza soluciones que ya han sido probadas en el pasado para problemas concretos. El dominio de los patrones de diseño es una práctica que se tiene que perfeccionar y practicar, es necesario conocer las ventajas y desventajas que ofrece cada uno de ellos, pero sobre todo requiere de experiencia para identificar dónde se deben utilizar.

Lo más importante de los patrones de diseño es que evita tener que reinventar la rueda, ya que son escenarios identificados y su solución está documentada y probada por lo que no es necesario comprobar su efectividad. Además, los patrones de diseño se basan en las mejores prácticas de programación.

Esta asignatura aporta al perfil del Ingeniero en Informática los conocimientos y habilidades necesarias para el desarrollo de aplicaciones empresariales bajo una arquitectura sólida que sea robusta, escalable y fácil de mantener.

1 Sistema de Asignación y Transferencia de Créditos Académicos

Intención didáctica
El docente deberá promover en el estudiante el desarrollo de habilidades que le permitan implementar patrones de diseño apropiadas al contexto y las características del módulo o componente del software. La asignatura está dividida en tres unidades en las cuales se abordan los temas según las tres categorías de patrones de diseño: Patrones Creacionales, Patrones Estructurales y Patrones de Comportamiento. La unidad uno se enfoca en los patrones creacionales y contempla los siguientes patrones: Patrón Factory Method, Patrón Abstract Factory, Patrón Singleton, Patrón Builder, Patrón Prototype y Patrón Object Pool. La unidad dos se enfoca en los patrones estructurales y contempla los siguientes patrones: Patrón Adapter, Patrón Bridge, Patrón Composite, Patrón Facade, Patrón Decorator, Patrón Proxy y Patrón FlyWeight. En la unidad tres se enfoca en los patrones de comportamiento y contempla los siguientes patrones: Iterator, Command, Observer, Template Method, Strategy, Chain of Responsability, Interpreter, Mediator, Memento, null Object, State y Visitor.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico del Valle de Oaxaca, 16 de octubre de 2019.	Academia de Ingeniería Informática e Ingeniería en Tecnologías de la Información y Comunicaciones del ITVO	Taller para generar la especialidad de la Ingeniería en Tecnologías de la Información y Comunicaciones.

4. Competencia(s) a desarrollar

Competencia específica de la asignatura
Desarrolla aplicaciones bajo una arquitectura sólida utilizando patrones de diseño.

5. Competencias previas

Modela software mediante diagramas UML (Diagrama de clases y secuencia).
Desarrolla programas usando paradigma Orientada a Objetos y programación funcional.

6. Temario

No.	Temas	Subtemas
1	Patrones creacionales	1.1 Factory Method 1.2 Abstract Factory 1.3 Singleton 1.4 Builder 1.5 Prototype 1.6 Object Pool
2	Visualización de la información	2.1 Adapter 2.2 Bridge 2.3 Composite 2.4 Facade 2.5 Decorator 2.6 Proxy 2.7 FlyWeight
3	Herramientas de Aprendizaje Máquina	3.1 Iterator 3.2 Command 3.3 Observer 3.4 Template Method 3.5 Strategy 3.6 Chain of Responsability 3.7 Interpreter 3.8 Mediator 3.9 Memento 3.10 Null Object

		3.11 State 3.12 Visitor

7. Actividades de aprendizaje de los temas

1. Patrones de diseño creacionales	
Competencias	**Actividades de aprendizaje**
Competencia Específica: Implementa patrones de diseño creacionales para crear objetos de manera controlada en una aplicación. **Competencias Genéricas:** Capacidad de análisis y síntesis. Capacidad de investigación. Capacidad de abstracción. Soluciona problemas.	Elaborar de forma individual el diagrama de clase y de secuencia que represente la forma de interacción de los componentes de cada uno de los patrones de diseño (Factory Method, Abstract Factory, Singleton, Builder, Prototype y Object Pool); para la elaboración del diagrama se puede utilizar cualquiera de las siguientes herramientas (día, umbrello, Microsoft visio, Enterprise Architect u otros). Implementar en algún lenguaje de programación (Java, C#, PHP, Kotlin o algún otro lenguaje con soporte de orientación a objetos o funcional) cada uno de los patrones de diseño (Factory Method, Abstract Factory, Singleton, Builder, Prototype y Object Pool).
2. Patrones de diseño estructurales	

Competencias	Actividades de aprendizaje
Competencia Específica: Implementa patrones de diseño estructurales para formar estructuras más grandes y complejas entre objetos y las relaciones entre clases. **Competencias Genéricas**: Capacidad de análisis y síntesis. Capacidad de investigación. Capacidad de abstracción. Soluciona problemas.	Elaborar de forma individual el diagrama de clase y de secuencia que represente la forma de interacción de los componentes de cada uno de los patrones de diseño (Adapter, Bridge, Composite, Facade, Decorator, Proxy y FlyWeight); para la elaboración del diagrama se puede utilizar cualquiera de las siguientes herramientas (día, umbrello, Microsoft visio, Enterprise Architect u otros). Implementar en algún lenguaje de programación (Java, C#, PHP, Kotlin o algún otro lenguaje con soporte de orientación a objetos o funcional) cada uno de los patrones de diseño (Adapter, Bridge, Composite, Facade, Decorator, Proxy y FlyWeight).
3. Patrones de comportamiento	
Competencias	**Actividades de aprendizaje**
Competencia Específica: Implementa patrones de diseño de comportamiento para simplificar la forma en que los objetos se comunican e interactúan entre sí . **Competencias Genéricas:** Aplica actividades de búsqueda, selección y análisis de información en distintas fuentes. Resolución de problemas. Presentación de resultados.	Elaborar de forma individual el diagrama de clase y de secuencia que represente la forma de interacción de los componentes de cada uno de los patrones de diseño (Iterator, Comand, Observer, Template Method, Strategy, Chain of Responsability, Interpreter, Mediator, Memento, Null Object, State y Visitor); para la elaboración del diagrama se puede utilizar cualquiera de las siguientes herramientas (día, umbrello, Microsoft visio, Enterprise Architect u otros) Implementar en algún lenguaje de programación (Java, C#, PHP, Kotlin o algún otro lenguaje con soporte de orientación a objetos o funcional) cada uno de los patrones de diseño (Iterator, Comand, Observer, Template Method, Strategy, Chain of

	Responsability, Interpreter, Mediator, Memento, Null Object, State y Visitor).

8. Práctica(s)

Practica 1. Implementar el patrón singletón combinado con Object Pool para la conexión a un origen de datos de un proyecto con acceso a datos.

Práctica 2. Del proyecto anterior aplicar al menos 3 patrones de diseño de la categoría estructural para reforzar los aprendizajes adquiridos.

Práctica 3. Del proyecto anterior aplicar al menos 3 patrones de diseño de la categoría comportamental para que el proyecto finalmente quede con una estructura sólida y con lo facilidad de realizar los cambios entre capas con la implementación de los patrones de comportamiento.

9. Proyecto de asignatura

La intención del proyecto que plantee el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual y legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.

- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.

- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y especificas a desarrollar.

- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de "evaluación para la mejora continua", la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Realizar prácticas con herramientas de limpieza de datos.

Realizar prácticas con herramientas de extracción de características.

Examen escrito.

Proyecto integrador que ponga en práctica los conocimientos obtenidos en esta asignatura.

Participación en eventos académicos, tales como Congresos o concursos donde se traten los avances en el área de análisis inteligente de datos.

Investigación documental.

Rúbricas para la evaluación de las prácticas.

Utilizar herramientas de desarrollo y programación.

11. Fuentes de información

- Blancarte Oscar Javier. (2016). Introducción a los patrones de diseño. Un enfoque práctico. México
- Elisabeth Freeman, Kathy Sierra. (2004). Head First Design Pattern. O' Reilly
- Mark Richards. (2015). Software Architecture Pattern. O' Reilly
- Soshin Alexey. (2018). Design Patterns with kotlin. Packt Publishing.