



<p>DISEÑO DE BASES DE DATOS</p> <p>Año 2020</p>	<p>Carrera: Maestría en Ingeniería de Software</p> <p>Docentes Responsables: Dra. Silvia Gordillo Mg. Javier Bazzocco, Mg. Alejandra Lliteras</p> <p>Duración: 108 hs.</p>
--	---

OBJETIVOS GENERALES:

- Discutir desde el punto de vista de la arquitectura de software, la problemática relacionada con el diseño y la implementación de sistemas orientados a objetos con persistencia en diferentes tipos de base de datos.
- Analizar las nuevas herramientas, tecnologías y paradigmas para almacenar y recuperar eficientemente la información generada.

CONTENIDOS MINIMOS:

- Persistencia Orientada a Objetos
- Alternativas de implementación
- Patrones de diseño utilizados en Persistencia
- Frameworks y productos

PROGRAMA

-Persistencia Orientada a Objetos

- Principios de transparencia de la persistencia.
 - Requisitos.
 - Beneficios a nivel diseño y a nivel implementación.
 - Ortogonalidad de persistencia
- Persistencia por alcance
 - Concepto.
 - Ventajas y consecuencias de su aplicación.
 - Implementación.
- Transacciones
 - Manejo de transacciones bajo el paradigma OO.
 - Esquemas pesimista y optimista.



- Demarcación de transacciones.
- Transacciones explícitas vs implícitas.
- Versionamiento de instancias para control de concurrencia.
- Operaciones CRUD
 - Diferencias entre esquema tradicional y a partir de persistencia por alcance.
 - Optimización de performance

- Alternativas de implementación

- Mapeo Objeto-Relacional
 - Cualidades deseables de la integración
 - Performance sin compromiso
 - Transparencia.
 - Soporte para múltiples bases de datos.
 - Independencia del mapeador.
 - Estrategias de mapeo
 - Mapeo de los principales elementos del paradigma orientado a objetos
 - Mapeo de colaboradores
 - Mapeo de jerarquías
 - Optimización
 - Diferentes tipos de colecciones.
 - Mapeo de conjuntos y listas.
 - Lenguaje de consulta HQL
 - Introducción.
 - Principales elementos.
 - Path expressions.
 - Subconsultas.
 - Joins.
- Bases de datos NOSQL
 - Introducción
 - Comparación con las bases de datos relacionales
 - Diferentes tipos de bases de datos
 - Orientadas a documentos
 - Familias de columnas
 - Clave valor
 - Orientadas a grafos
 - Performance
 - MONGODB
 - Esquema de trabajo
 - Principales conceptos
 - Colecciones
 - Documentos



- Réplicas
 - Consultas
 - Sharding

-Patrones de diseño utilizados en Persistencia

- Proxy: utilización del Proxy para optimización de performance. Lazy loading.
- Decorator: intercepción de mensajes para demarcación de transacciones.
- Repository: implementación de repositories para acceder eficientemente al modelo.
- Root Object: representación del sistema a través de clases dedicadas.
- DTO: transferencia de información entre las diferentes capas de la aplicación.
- DAO y razones para no utilizarlo: principios de diseño para capa de acceso a Datos. Consecuencias indeseadas de la aplicación de este patrón. Modelos anémicos.

-Frameworks y productos

- Spring framework.
 - Inversión de control.
 - Inyección de dependencias
 - Configuración.
 - Springboot
- Hibernate
 - Configuración.
 - Ejemplos de uso.
- MONGODB
 - Instalación
 - Configuración
 - Pruebas.
- ELASTIC SEARCH
 - Instalación
 - Configuración
 - Pruebas

ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACIÓN

Durante el curso, los alumnos llevarán adelante diferentes trabajos de experimentación que les permitirá poner en práctica los conceptos visitados durante las clases, así como emplear los frameworks y productos presentados. Adicionalmente, el alumno realizará relevamiento y análisis bibliográfico en relación a los temas del programa, lo que le permitirá sustentar en base al mismo, el informe de su trabajo final.

METODOLOGIA DE EVALUACION

La evaluación se lleva a cabo a través de la presentación de un trabajo de diseño, implementación y justificación teórica de un ejercicio que se plantea al final de la cursada en relación a los temas abordados en el programa.

Dicho trabajo podrá realizarse en grupos de 2 personas utilizando diferentes plataformas tecnológicas, las cuales serán acordadas con el docente.

El trabajo se divide en cinco etapas. En la primera etapa, se debe presentar una línea base de requerimientos, la cual debe ser aprobada por el docente. La segunda etapa, consiste en presentar un diseño orientado a objetos, en el que al menos, se debe presentar un diseño de clases y de interacción para las principales funciones consideradas en la propuesta de la etapa 1. La tercera etapa, comprende la implementación de una parte del sistema descrito en las etapas anteriores, con su correspondiente persistencia de la información en una base de datos a establecer de común acuerdo. En la cuarta etapa, se deberá redactar un documento donde se justifique, mediante un trabajo de relevamiento y análisis de bibliografía, los conceptos teóricos y las tecnologías empleadas durante el trabajo propuesto hasta la etapa anterior. Por último, la quinta etapa, considera un coloquio individual del trabajo presentado.

Cada etapa del trabajo debe ser aprobada por el docente para avanzar a la etapa siguiente. En caso de desaprobación de una etapa, se podrá realizar una reentrega que contemple las correcciones indicadas por el docente. El trabajo se considera aprobado, cuando sus cinco etapas han sido aprobadas.

El tiempo máximo para realizar el trabajo desde el inicio del mismo, es de 6 meses.

En caso de desaprobación del trabajo del curso, el alumno deberá recursarlo.

COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.5- Diseñar y gestionar el almacenamiento de datos, aplicando diferentes tecnologías y frameworks de persistencia de datos actuales.

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

BIBLIOGRAFÍA BASICA

- Atzeni, P., Bugiotti, F., Cabibbo, L., & Torlone, R. (2020). Data modeling in the NoSQL



world. *Computer Standards & Interfaces*, 67, 103149.

- Fisher, P., & Murphy, B. D. (2016). *Spring Persistence with Hibernate*. Apress.
- Juneau, J. (2020). *Java Persistence Query Language*. In *Jakarta EE Recipes* (pp. 527-567). Apress, Berkeley, CA.
- Keith, M., Schincariol, M., & Nardone, M. (2018). *Pro JPA 2 in Java EE 8: An In-Depth Guide to Java Persistence APIs*. Apress.
- Malhotra, R. (2019). *Rapid Java Persistence and Microservices: Persistence Made Easy Using Java EE8, JPA and Spring*. Apress.
- Meier, A., & Kaufmann, M. (2019). *Nosql databases*. In *SQL & NoSQL Databases* (pp. 201-218). Springer Vieweg, Wiesbaden.
- Milhalcea, V. (2016) *High-Performance Java Persistence* octubre de 2016 -ISBN-10: 973022823X
- Raj, P., & Deka, G. C. (2018). *A Deep Dive into NoSQL Databases: The Use Cases and Applications*. Academic Press.

**BIBLIOGRAFÍA COMPLEMENTARIA**

- Acharya, B., Jena, A. K., Chatterjee, J. M., Kumar, R., & Le, D. N. (2019). NoSQL Database Classification: New Era of Databases for Big Data. *International Journal of Knowledge-Based Organizations (IJKBO)*, 9(1), 50-65.
- Aniche, M., Yoder, J., & Kon, F. (2019, May). Current challenges in practical object-oriented software design. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)* (pp. 113-116). IEEE.
- Ansari, M.H., Tabatab Vakili, V. & Bahrak, B. Evaluation of big data frameworks for analysis of smart grids. *J Big Data* 6, 109 (2019). <https://doi.org/10.1186/s40537-019-0270-8>
- Cutler, J., & Dickenson, M. (2020). NoSQL Databases. In *Computational Frameworks for Political and Social Research with Python* (pp. 117-126). Springer, Cham.
- Dasadia, C. (2017). *MongoDB Administrator's Guide: Over 100 practical recipes to efficiently maintain and administer your MongoDB solution*. Packt Publishing Ltd.
- Fisher, P., & Murphy, B. D. (2016). Architecting Your Application with Spring, Hibernate, and Patterns. In *Spring Persistence with Hibernate* (pp. 1-16). Apress, Berkeley, CA.
- Mittal, M., Balas, V. E., & Hemanth, D. J. (Eds.). (2018). *Data Intensive Computing Applications for Big Data (Vol. 29)*. IOS Press.
- Parmar, R. R., & Roy, S. (2018). MongoDB as an Efficient Graph Database: An Application of Document Oriented NOSQL Database. *Data Intensive Computing Applications for Big Data*, 29, 331.
- Rajput, D. (2017). *Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more*. Packt Publishing Ltd.
- Reniers, V., Van Landuyt, D., Rafique, A., & Joosen, W. (2019). Object to NoSQL Database Mappers (ONDM): A systematic survey and comparison of frameworks. *Information Systems*.
- Shin, K., Hwang, C., & Jung, H. (2017). NoSQL database design using UML conceptual data model based on Peter Chen's framework. *International Journal of Applied Engineering Research*, 12(5), 632-636.



- Singh, K., Ianculescu, A., & Torje, L. P. (2018). Design Patterns and Best Practices in Java: A comprehensive guide to building smart and reusable code in Java. Packt Publishing Ltd.