**GA8- CODIFICACIÓN DE MÓDULOS DEL SOFTWARE SEGÚN REQUERIMIENTOS DEL PROYECTO**

**“220501096”**

**“AA2-EV01”**

**POR:**

**JUAN CAMILO MURGAS JACOME**

**CENTRO DE LA TECNOLOGÍA DEL DISEÑO Y LA PRODUCTIVIDAD EMPRESARIAL**

**REGIONAL CUNDINAMARCA – SENA – GIRARDOT**

**ANÁLISIS Y DESARROLLO DE SOFTWARE**

**FICHA 2977481**

**INSTRUCTOR: MILTON BARBOSA**

**8/30/2025**

**INTRODUCCIÒN**

El desarrollo de aplicaciones orientadas a la gestión de información requiere la implementación de módulos que permitan la interacción entre el software y las bases de datos. En este contexto, el presente trabajo aborda la construcción de un módulo de proyecto en Java utilizando JDBC (Java Database Connectivity) como tecnología principal para el manejo de datos.

El proceso se fundamenta en los artefactos previamente elaborados durante el ciclo de vida del software, tales como el diagrama de clases, diagramas de casos de uso, historias de usuario, prototipos y plan de trabajo técnico. Asimismo, se incorporan prácticas de versionamiento y estándares de codificación que garantizan la calidad, legibilidad y escalabilidad del sistema.

El módulo desarrollado permite ejecutar las operaciones básicas de persistencia de datos: inserción, consulta, actualización y eliminación (CRUD) sobre la base de datos, asegurando una comunicación eficiente entre la aplicación y el sistema gestor de base de datos.

**OBJETIVOS**

**Objetivo General:**

Desarrollar un módulo de software en Java que gestione la interacción con bases de datos a través de JDBC, aplicando los estándares de codificación y los artefactos del ciclo de vida del software previamente definidos.

**Objetivos Específicos:**

* Implementar la conexión entre la aplicación Java y la base de datos utilizando JDBC.
* Desarrollar funcionalidades de inserción, consulta, actualización y eliminación de datos (CRUD).
* Aplicar estándares de codificación en la definición de clases, métodos, variables y paquetes.
* Integrar los artefactos diseñados en fases anteriores (casos de uso, diagramas, prototipos, historias de usuario) en la implementación del módulo.
* Utilizar herramientas de versionamiento para controlar el ciclo de desarrollo y la trazabilidad del código fuente.

**METODOLOGÍA**

El desarrollo del módulo en Java con conexión a bases de datos mediante JDBC se llevó a cabo siguiendo un enfoque estructurado que integra las fases del ciclo de vida del software y las buenas prácticas de codificación. La metodología aplicada comprendió los siguientes pasos:

**1. REVISIÓN DE ARTEFACTOS PREVIOS**

Se revisaron los entregables elaborados en fases anteriores: diagrama de clases, diagramas de casos de uso, historias de usuario, diseños y prototipos. Estos documentos permitieron establecer los requisitos funcionales y técnicos necesarios para la construcción del módulo.

**2. DISEÑO DEL MÓDULO Y ESTRUCTURA DEL PROYECTO**

Se definió la arquitectura del módulo, organizando el proyecto en paquetes Java con un esquema lógico que facilita el mantenimiento y escalabilidad. Se establecieron convenciones de nombramiento para clases, métodos y variables, en cumplimiento con los estándares de codificación.

**3. IMPLEMENTACIÓN DE LA CONEXIÓN A LA BASE DE DATOS**

Se configuró la conexión a la base de datos MySQL utilizando JDBC. Para ello se empleó la clase DriverManager y un objeto Connection que permite establecer la comunicación entre la aplicación y el motor de base de datos.

**4. CODIFICACIÓN DE FUNCIONALIDADES CRUD**

Se implementaron las operaciones fundamentales de persistencia de datos:

* **Inserción:** registros de nuevos usuarios en la tabla.
* **Consulta**: recuperación y visualización de datos almacenados.
* **Actualización:** modificación de información existente.
* **Eliminación**: supresión de registros específicos.

Estas operaciones fueron desarrolladas con objetos Statement y PreparedStatement para garantizar seguridad y eficiencia en las consultas.

**5. CONTROL DE VERSIONES DEL CÓDIGO**

Se utilizó una herramienta de control de versiones (Git) para gestionar los cambios en el código fuente, permitiendo un mejor seguimiento de la evolución del proyecto y la trazabilidad de modificaciones.

**6. PRUEBAS DE FUNCIONAMIENTO**

Se realizaron pruebas unitarias y de integración para validar la correcta ejecución de las operaciones CRUD y la estabilidad de la conexión a la base de datos. Además, se verificó que las consultas manejaran adecuadamente posibles errores como claves duplicadas, registros inexistentes o campos nulos.

**7. DOCUMENTACIÓN Y ENTREGA**

Finalmente, se documentó el proceso de codificación, la estructura del proyecto y los resultados obtenidos, consolidando la evidencia como soporte del desarrollo realizado.

**CONCLUSIÒN**

La construcción del módulo en Java con JDBC permitió fortalecer las competencias en desarrollo de aplicaciones de escritorio que interactúan con sistemas gestores de bases de datos. A partir de la codificación de operaciones CRUD, se logró materializar los artefactos diseñados en etapas anteriores del proyecto, evidenciando la importancia de una planificación estructurada.

El cumplimiento de estándares de codificación, el uso de herramientas de versionamiento y la aplicación de buenas prácticas en la arquitectura del software contribuyeron a obtener un sistema más robusto, escalable y mantenible. De esta manera, se consolidan las bases necesarias para el desarrollo de soluciones tecnológicas aplicables a entornos reales en el ámbito académico y profesional.

**BIBLIOGRAFIAS**

* Deitel, H. & Deitel, P. (2017). *Java: How to Program*. Pearson Education.
* Oracle. (2024). *JDBC™ API Documentation*. Disponible en: https://docs.oracle.com/javase/
* Sierra, K. & Bates, B. (2019). *Head First Java*. O’Reilly Media.
* Sommerville, I. (2016). *Ingeniería del Software*. Pearson Educación.
* Welling, L., & Thomson, L. (2018). *MySQL Tutorial*. Addison-Wesley.