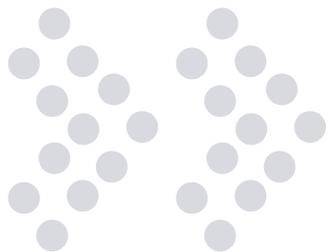




PROGRAMA DE ESTUDIOS

Java Hibernate
Domina el Framework de Java



CONTENIDO



Aprende el framework más utilizado del mercado JAVA para trabajar con Base de Datos, y convierte tus aplicaciones JAVA en multiplataforma y multibase de datos. Automatiza las operaciones con la base de datos, aumentando notablemente la productividad y disminuyendo hasta un 70% el tiempo de codificación de la capa de acceso a datos.

REQUISITOS

Se requieren conocimientos sólidos en **Java** o haber realizado el curso Java *J2SE Standard Programming*.

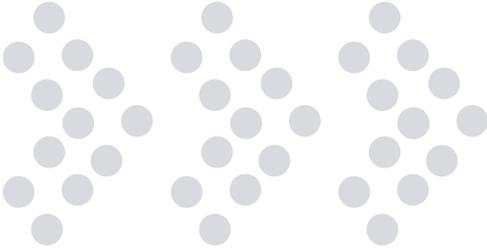
MODALIDADES

Puedes tomar este curso en modalidad **presencial** o modalidad **Online** - en vivo.

DURACIÓN

Este curso tiene una duración de **12 horas** académicas.





EN ESTE CURSO APRENDERÁS A



- ❖ Conocer las clases principales para implementar Hibernate y la funciones de cada una.
- ❖ Resolver la diferencia entre el modelo relacional y de objetos.
- ❖ Realizar acciones básicas de insert, delete, update y select con Hibernate.
- ❖ Ejecutar Queries HQL y SQL desde Java.
- ❖ Utilizar y configurar Apache log4j.
- ❖ Utilizar queries parametrizadas.
- ❖ Aplicar técnicas de paginado con Hibernate.
- ❖ Utilizar queries guardados en archivos de texto desde Hibernate.
- ❖ Utilizar las diferentes soluciones para mapear herencia con Hibernate.
- ❖ Conocer la utilización de las diferentes propiedades en el mapeo de asociaciones.
- ❖ Manejar transacciones con Hibernate.
- ❖ Conocer herramientas complementarias que permiten automatizar el proceso de generación de código.





TEMARIO



1. Persistencia de objetos

¿Qué es la persistencia?

- Definición
- Bases de datos relacionales
- Archivos planos

Modelo relacional vs. Modelo oo

- Problemática
- Una tabla, una clase
- El problema de la herencia
- El problema de la identidad
- El problema de las asociaciones
- El problema de la navegación

¿Qué es orm?

- Definición

Organización

- Ventajas

Tecnologías orm

- User-defined daos
- Ejb entity beans
- Jpa (java persistence api)
- Hibernate

La necesidad de una dal (data access layer)

- Arquitectura multicapa (n-tier architecture)
- ¿Qué es dal?
- ¿Qué es pojo?
- Ventajas de una arquitectura multi-capa

2. Configuración de hibernate

Introducción

Jars necesarios

- Archivos binarios
- Hibernate3.jar
- Otros .jar necesarios
- Jdbc driver .jar

Links

Documentación



3. Introducción a un proyecto con hibernate

Configuración

- Jerarquía de directorios
- La conexión con la base de datos

Propiedades de hibernate

- Introducción
- Hibernate.connection.driver_class
- Hibernate.connection.url
- Hibernate.connection.username
- Hibernate.connection.password
- Hibernate.dialect
- Los dialectos disponibles
- Ejemplo el archivo hibernate.properties

Configuración del rdbms

- MySQL como rdbms
- Creación de base de datos y tablas

Creación de un pojo

- ¿Qué es un pojo?
- Características de un pojo
- Ejemplo de un pojo

Creación de un xml mapper

- ¿Qué es?
- El archivo .hbm.xml
- El tag

Creación de una clase tester

- La clase testautos
- La interfaz org.hibernate.session
- Registración de propiedades y mappers
- ¿Cómo realizar una inserción?
- ¿Cómo realizar una consulta?
- ¿Cómo realizar una actualización?
- ¿Cómo realizar una eliminación?

4. Interfaces / loggins / sesiones

La clase org.hibernate.cfg.configuration

La interfaz org.hibernate.sessionfactory

La interfaz org.hibernate.session

La interfaz org.hibernate.query

La interfaz org.hibernate.transaction

El paquete org.hibernate.classic

Logging

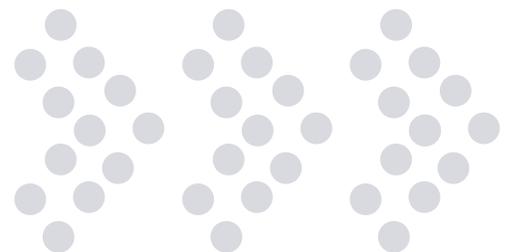
¿Qué es?

El archivo log4j.properties

La propiedad hibernate.show_SQL

Administración de sesiones

- ¿Qué es un singleton?
- ¿Por qué utilizar un singleton?
- Utilización



5. Hql: hibernate query language

¿Qué es?

¿Por qué utilizarlo?

Cláusulas

- From
- Select
- Where
- Group by
- Order by

Utilización de funciones

- Introduccion
- Count
- Sum
- Avg
- Max
- Min

Consultas SQL nativas

- ¿Qué es una consulta nativa?
- El método createSQLquery()

Valores por parámetro

Paginación

- Definición
- El método setmaxresults()
- El método setfirstresult()

Named queries

- ¿Qué es un named query?
- El named query mapping
- El método getnamedquery()

6. Mapeo de herencia de clases

Problemática

Soluciones

- Una tabla por clase concreta
- Una tabla por subclase
- Una tabla por jerarquía de clases

7. Transacciones

La interfaz org.hibernate.transaction

Que significa transparent write behind

El metodo load()

El metodo get()

El método begintransaction()

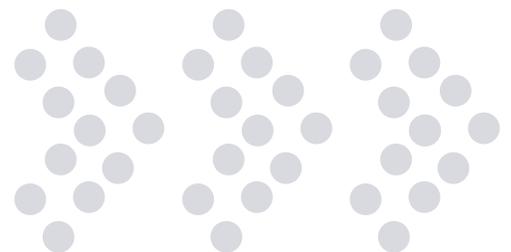
El método flush()

El método setflushmode()

El método commit()

El método rollback()

Utilización de rollback() y commit()



8. Asociaciones

One-to-many/many to one

- Mapped by
- Cascading
- Fetching

One-to-one

Many to many

9. Herramientas complementarias

Introducción

Andromda

Xdoclet

Schemaexport (hbm2ddl)

Middlegen

Codegenerator (hbm2java)

10. Laboratorios

Hola mundo con hibernate

Persistiendo un concesionario

Manejando hql

Queries parametrizadas

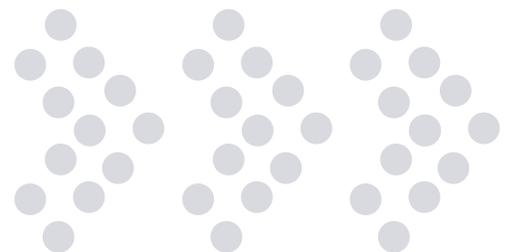
Paginación

Namedqueries

Manejo de herencia

Asociaciones

Proyecto integrador



¿POR QUÉ EN CEGOS?



Testimonios de nuestros alumnos



Un alumno por PC



Certificación Internacional



Clientes que confían en nosotros



Grupos reducidos y personalizados



Garantía de aprendizaje



Cursos con modalidad taller



Bolsa de empleo

MODALIDAD PRESENCIAL

- Acceso a Wifi de alta velocidad
- Grupos reducidos y prácticos
- Un computador por persona
- Material de apoyo digital
- Certificación

MODALIDAD ONLINE

- Grabación de las clases ejecutadas
- Grupos reducidos y prácticos
- Asistencia administrativa
- Clase en tiempo real por Videoconferencia
- Consultas en vivo
- Certificación

CERTIFICACIÓN

Los mismos contienen:

- Rubrica de autoridades competentes
- Datos personales del alumno
- Horas lectivas invertidas
- Plan de estudios
- Nota final

Centro autorizado por:



CEGOS

Conocimiento para el desarrollo

Sopocachi, Av. Sánchez Lima, Edif. Valentina, Nro. 2326,
Oficina "B",
(591) 68223703 / (591) 2 - 2420172
www.cegos.com.bo - consultas@cegos.com.bo

