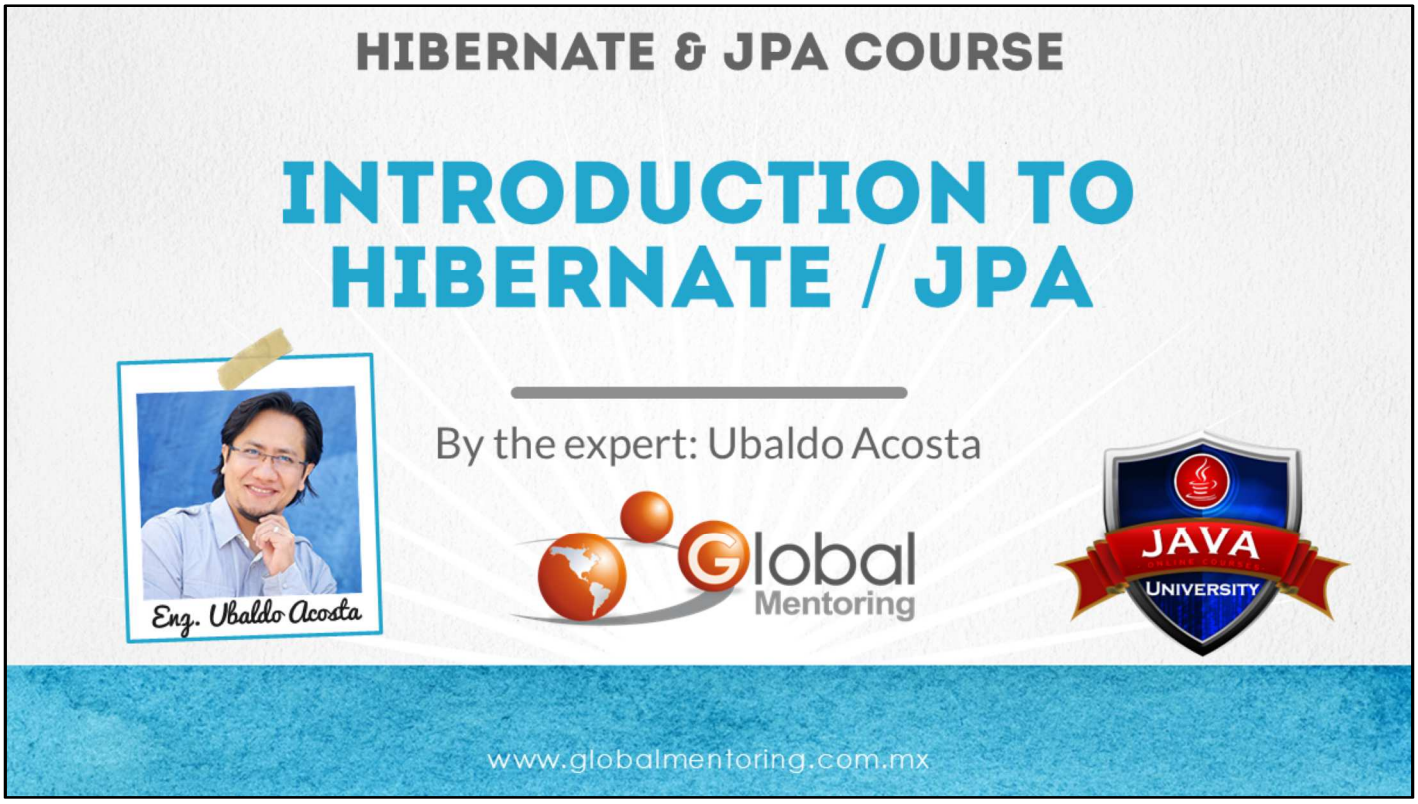


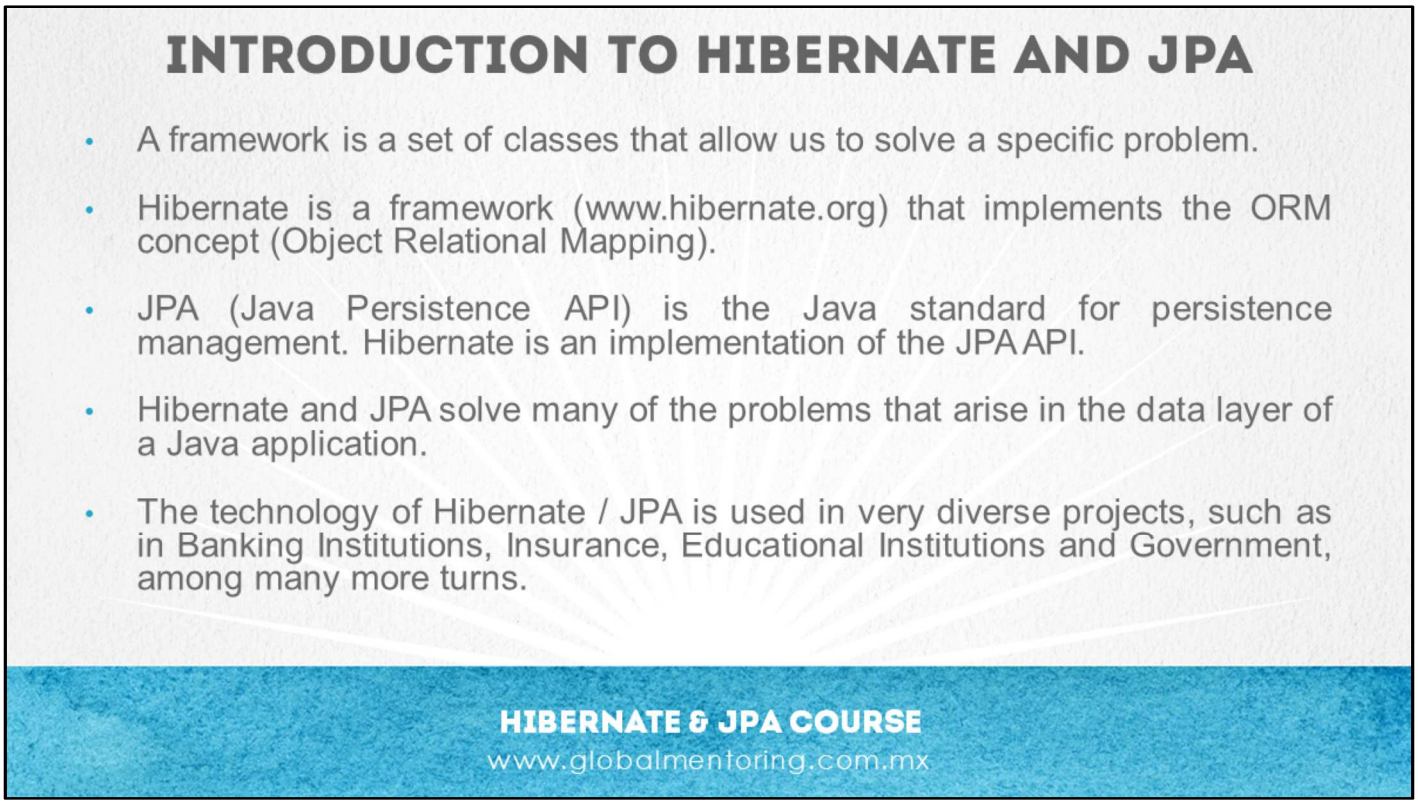
Hello, Ubaldo Acosta greets you. Welcome to this Hibernate and JPA course.

In this first lesson we are going to review an introduction to the Hibernate framework, as well as the technology of JPA (Java Persistence API).

We will also mention the need to learn to use correctly and once and for all this Java persistence technology, since it is one of the most used technologies around the world.

In this course of Hibernate / JPA we will study the technologies that will allow us to create Java applications that need to communicate with a database in a more efficient and simple way than using JDBC. Through practical examples we will review the characteristics of this technology.

So if you're ready, we will too. Let's start immediately.



In this first lesson, we will review an Introduction to the Hibernate and JPA framework.

A Framework is a set of classes that allow us to solve a specific problem. In the case of the Hibernate Framework, the concept that will be implemented is the concept of ORM (Object-relational Mapping), later we will review in more detail what this concept consists of.

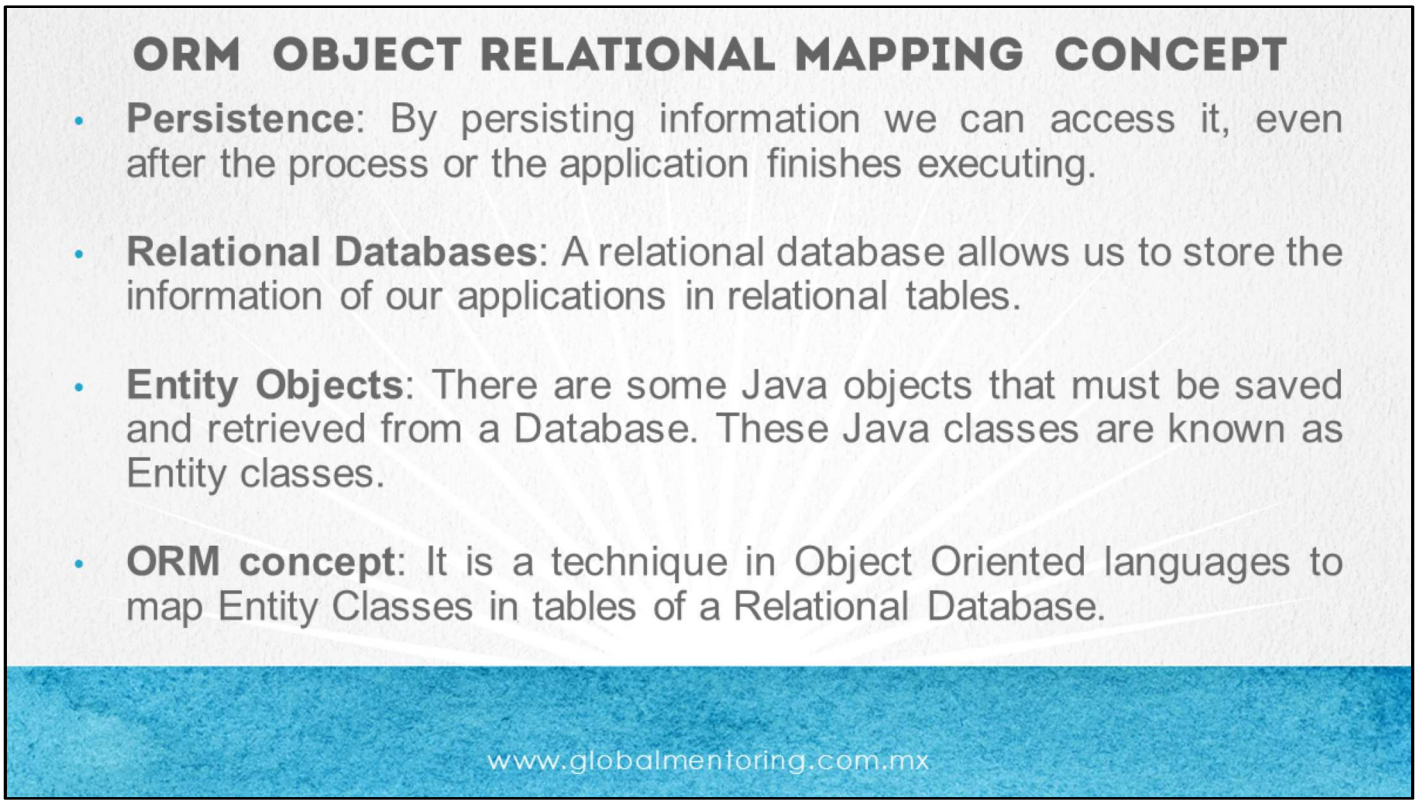
Hibernate is a framework (www.hibernate.org) that implements the ORM concept (Object Relational Mapping).

JPA (Java Persistence API) is the Java standard for persistence management. Hibernate is an implementation of the JPA API. Hibernate was one of the technologies that contributed the most to what JPA is to date. This Java standard would not be what it is without all the development that Hibernate had, and today Hibernate is one of the most popular implementations of JPA, and although there are other implementations in this course we will not focus on the use of JPA and Hibernate. due to the advantages that this union brings.

Hibernate and JPA solve many of the problems that arise in the data layer of a Java application. Hibernate can be used in an isolated manner, and JPA can be used with another implementation other than Hibernate, however we will study it together and it is already the student's decision to use these technologies in isolation or with another implementation.

The technology of Hibernate / JPA is used in very diverse projects, such as in Banking Institutions, Insurance, Educational Institutions and Government, among many more turns.

Of course you can also use this technology for your own projects, if you have the need to save and retrieve information from a database more efficiently, robustly and scalable, using the power of Java and now applying Hibernate and JPA technology.



Let's review the concept of ORM (Relational Mapping Object). The persistence of information is one of the most critical issues in a software application, in object-oriented languages such as Java, it is common to apply advanced persistence techniques such as the concept of ORM (Relational Mapping Object) or relational mapping model -object.

The ORM model allows that from a database model we can define Java classes that represent, in the simplest case, a relational database table. It should be mentioned that the mapping is not necessarily one to one, that is, there could be tables of databases that do not have a representation of a class in Java and vice versa, these techniques will be studied later.

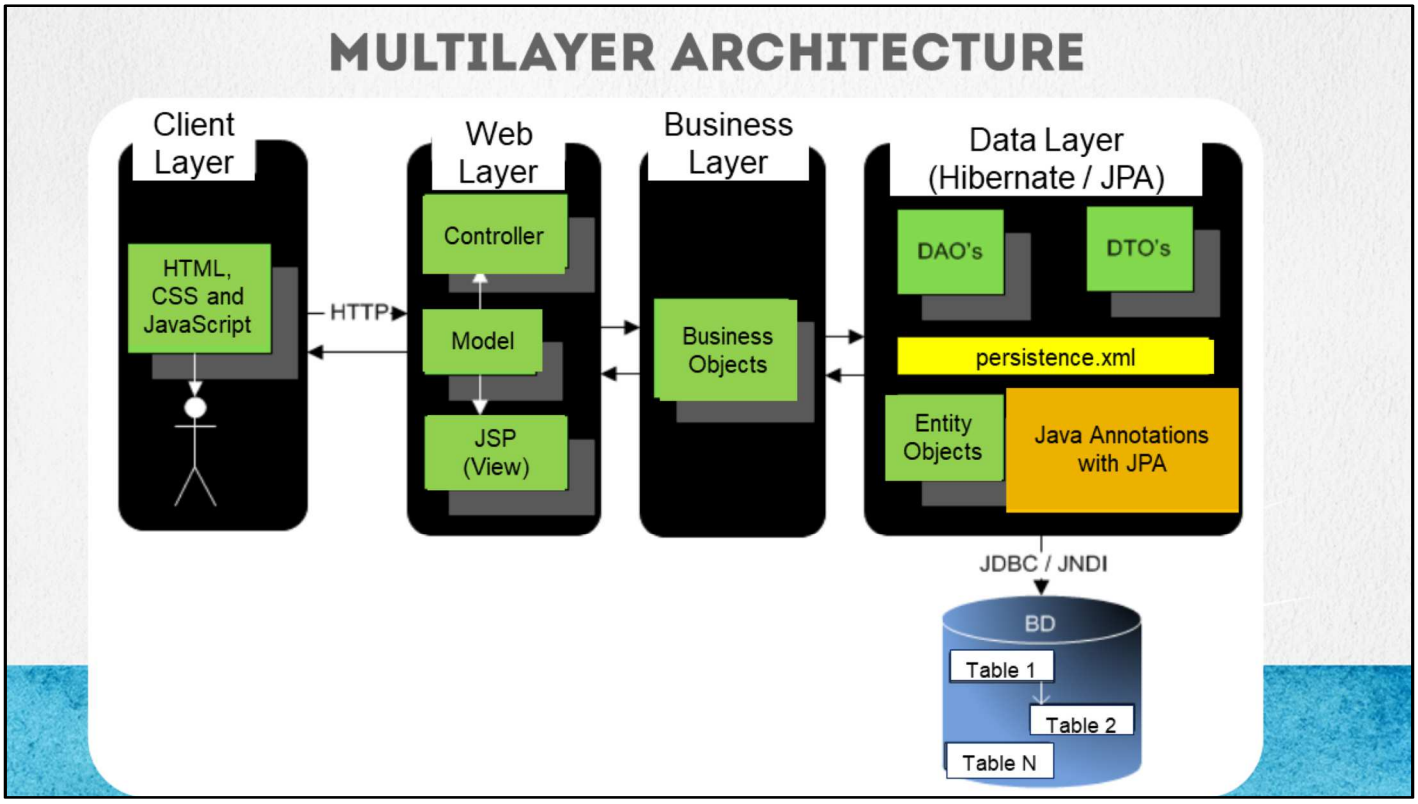
Let's review the concept of persistence. When we are talking about the concept of persistence, it means that the information that we are storing will be able to be accessed even after the process or application that we are using has finished executing.

In this course we are going to review that to persist information we are going to use a relational base, but we could also use other tools such as files or other types of databases that are not necessarily relational. In our case, a relational database will allow us to save information about our application or applications, and this information will persist in relational tables.

Let's review what we refer to with Entity Objects: In our applications there are some objects in Java that must be saved and retrieved from a database. These types of classes are known as Entity Classes or also known as Domain Classes, therefore, we can say that not all the objects of our application will be stored in a database.

Finally, we will review the concept of ORM. ORM is a technique in object-oriented languages for mapping Entity Classes into tables of a relational database. This means that there are some classes in Java that need to be saved in a database in order to access this information later.

In this course, when studying Hibernate and JPA technology, we will study how to map Java objects in relational database tables using these technologies.



Next we can observe a multi-layer architecture. This architecture is oriented to a Web application with Java, because it is the type of applications to which we are going to focus also to this course, but with this same knowledge you can create any type of application, whether Desktop, Web or Business applications. .

We can see that we have the client layer and we have a web layer. In the web layer we can use technologies such as JSPs, Struts, JSF, Spring MVC or any other type of Frameworks created to develop the presentation layer in a Java application. In the business layer we can use technology such as EJBs or the Spring framework. And in the Data Layer, which we will focus on this course, we can use technology such as JDBC or the Hibernate and JPA framework.

There are several concepts that we are going to comment on as we go forward, but we will be using design patterns such as the DAO design pattern and the DTO design pattern and we will also study the configuration of the Hibernate and JPA Framework. There are two possible configurations for the Hibernate or JPA framework, one is to use XML files or we can use Java annotations to configure the Entity classes and make the mapping of our Entity classes with the database.

If we use Hibernate in isolation, the file called hibernate.cfg.xml is the most important, since there is the configuration of the Hibernate framework, for example it has the connection information to the database, the list of Entity classes, Among several other features. However, throughout the course we will use Hibernate in combination with the JPA API (Java Persistence API), since today is the way in which Java applications are most commonly created, and instead of using the hibernate.cfg.xml file, we will use the persistence.xml file, however it will be very similar in both cases to use one or another option, we will choose the option that we consider best for your Java projects today.

We can represent the entity classes with xml mapping files or directly with Java annotations within the Entity classes, as we have said. These entity objects are usually represented by a database table and we can see that we can have N number of tables that will represent each of the entity objects of our Java application. Not necessarily an entity object is going to have a related table and vice versa, this type of mappings are the techniques that we will be studying in this course.

Then we can see that a data layer using Hibernate / JPA will use several concepts, both design patterns and configuration of the same framework, as well as libraries (.jar files) that we must add to our application. In this course we are going to put into practice that type of architectures since they are the type of applications used in the Java labor market and therefore it is the kind of applications that will allow us to obtain the necessary experience for the real world Java.