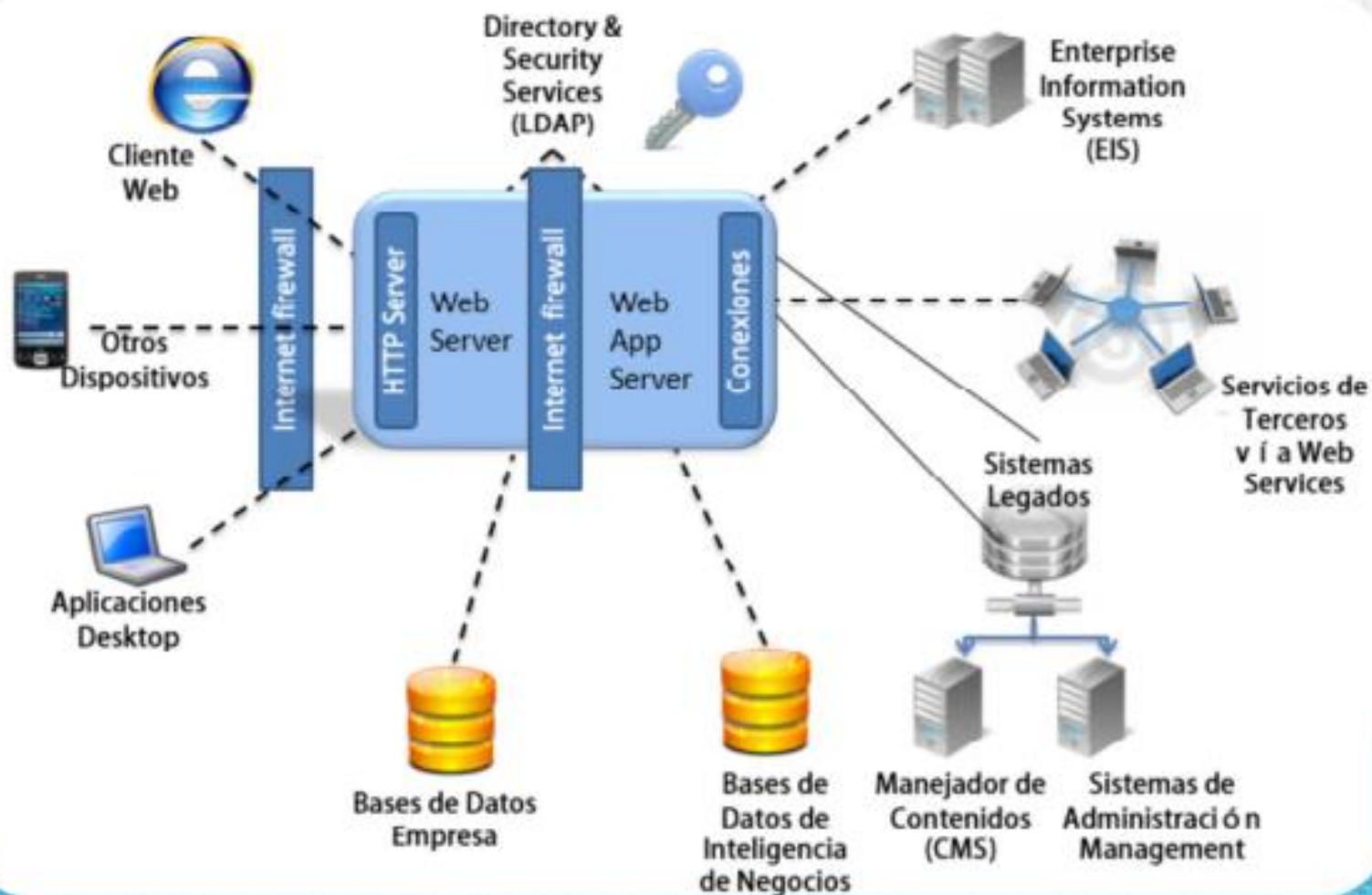


# ¿QUÉ ES UNA APLICACIÓN EMPRESARIAL?



Vivimos en un mundo globalizado, donde la eficiencia y productividad de las empresas es un factor crucial para el éxito o fracaso de las mismas. Los Sistemas de Información juegan un papel fundamental en la mejora y consolidación de las compañías.

En la figura podemos observar un ejemplo de configuración de una aplicación Empresarial. Java es la plataforma de desarrollo utilizada por excelencia para el desarrollo de aplicaciones empresariales a la medida. Con más de dos décadas en el mercado, este lenguaje de programación es uno de los más maduros, estables y confiables para el desarrollo de sistemas de misión crítica al día de hoy.

Java permite crear aplicaciones para usuarios de distintos tipos como son clientes de Escritorio, Web y Móviles. Las aplicaciones empresariales Java tienen a su cargo establecer las reglas de negocio de la empresa y/o sistema y ofrecer conectividad a los distintos tipos de clientes, con ello se logra ofrecer una solución integral a sus necesidades de sistemas de información a la medida.

La versión empresarial de Java (Java Enterprise Edition) cuenta con una enorme comunidad de programadores alrededor del mundo. A su vez, una de las mayores ventajas de Java es su independencia de plataforma, de esta manera, el programador puede seleccionar entre utilizar herramientas de paga o de software libre, y esto comienza desde el sistema operativo, hasta las herramientas de desarrollo, el servidor de aplicaciones, la base de datos, etc.

La versión empresarial de Java tiene como objetivo la simplificación en la programación para requerimientos empresariales, y con ello busca incrementar la productividad del desarrollador Java.

## ¿QUÉ ES JAVA EE?

- Un API es un conjunto de clases que nos permiten resolver una necesidad muy particular.
- Java EE es un conjunto de APIs enfocadas en brindar servicios empresariales, ej:
  - Transaccionalidad
  - Seguridad
  - Interoperabilidad
  - etc...
- Algunas de las APIs Empresariales de Java son:
  - JTA – Java Transaction API
  - JPA – Java Persistence API
  - JMS – Java Message Service
  - etc...

Para entender qué es Java EE, comencemos por responder a la pregunta de ¿Qué es un API?. Un API (Application Programming Interface) es un conjunto de clases que resuelven una necesidad muy particular. Por ejemplo el API de JDBC permite crear código Java para establecer la comunicación con una base de datos.

Java EE es un conjunto de API's enfocadas en brindar una serie de servicios que toda aplicación empresarial necesita, tales como: transaccionalidad, seguridad, interoperabilidad, persistencia, objetos distribuidos, entre muchos servicios más. Estas APIs se basan en un conjunto de especificaciones, las cuales pueden ser implementadas por empresas orientadas a software libre (Tomcat, Jboss, etc) o software comercial (Oracle, IBM, etc).

Algunas de estas APIs empresariales son:

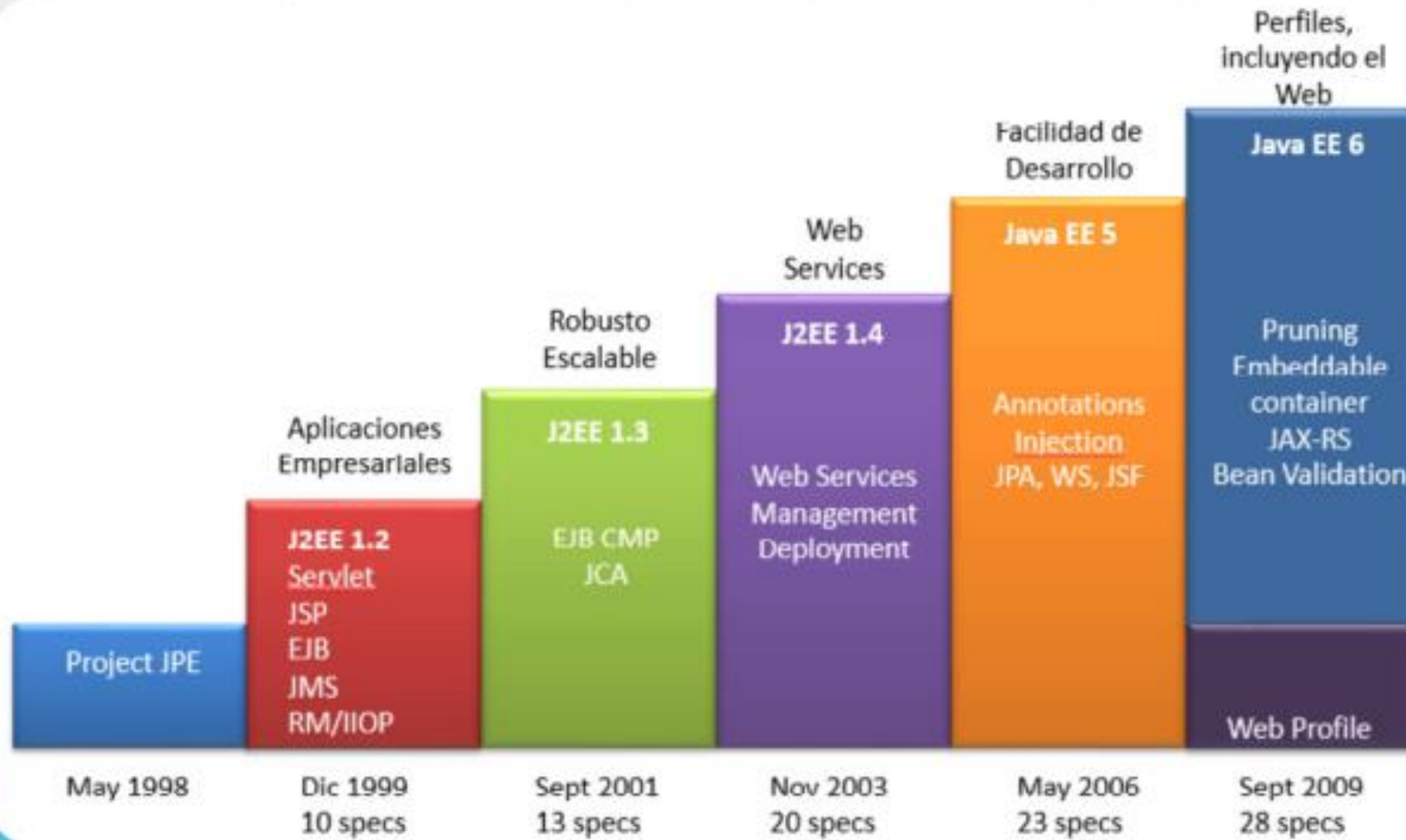
- Manejo de Transacciones: Java Transaction API (JTA)
- Persistencia: Java Persistence API (JPA)
- Mensajería: Java Message Service (JMS)
- Manejo de Servicios Web: Java API for XML Web Services (JAX-WS) y Java API for RESTful Web Services (JAX-RS)
- Seguridad: Java Authentication and Authorization Service (JAAS)
- Localización de objetos: Java Naming and Directory Interface (JNDI)
- Entre muchas APIs más

Una de las grandes ventajas de seleccionar estas tecnologías es que son el estándar propuesto por el grupo JCP (Java Community Process), el cual se encarga de revisar y liberar las especificaciones Java y las APIs empresariales respectivas.

---

En resumen, la versión empresarial de Java se puede entender como una extensión de la versión estándar (JSE), pero con la intención de facilitar el desarrollo de aplicaciones empresariales, permitiendo agregar de manera muy simple los servicios descritos anteriormente, y así crear aplicaciones Java robustas, poderosas, y de alta disponibilidad.

# HISTORIAL DE LAS VERSIONES EMPRESARIALES JAVA



En la figura podemos observar la historia de la versión empresarial de Java. En 1998 se liberó la primera versión y desde entonces ha evolucionado enormemente hasta lo que es al día de hoy.

Uno de los principales requerimientos en su primera versión fue el manejo de sistemas distribuidos, los cuales consisten en poder ejecutar componentes en distintos servidores. CORBA era la tecnología utilizada para cubrir este requerimiento.

Java, por su lado, propuso el protocolo RMI-IIOP (Remote Method Invocation–Internet Inter-ORB Protocol) para cubrir este tipo de requerimientos de llamadas distribuidas, y con este concepto en mente fue que liberó la primera versión de los Enterprise JavaBeans (EJBs). Además se introdujeron tecnologías como Servlets y JSPs para la creación de aplicaciones Web, y temas de mensajería con JMS.

En las siguientes versiones se lograron varios objetivos, tales como hacer las aplicaciones más robustas y escalables. Posteriormente se incluyó el tema de Web Services como parte del estándar.

La versión Java EE 5 dio un giro sustancial en la simplificación de la programación de aplicaciones empresariales. Esto como respuesta a que varios frameworks libres como Struts, Spring y Hibernate simplificaron la programación promoviendo el uso de clases puras de Java (POJO's). Además, el concepto de anotaciones permitió simplificar la configuración de clases como los EJB's, y se hizo opcional la configuración de estas clases y archivos xml.



# JAVA EE 7

CDI 1.1

Bean Validation 1.1

Interceptors 1.2

Concurrency 1.0

JSP JSTL

EL 3.0

JSF 2.2

JAX-RS 2.0

Servlet 3.1

Web Socket 1.0

JSON-P 1.0

Batch 1.0

JTA 1.2

EJB 3.2

JMS 2.0

JavaMail 1.5

JPA 2.1

JCA 1.7

Java EE 7



En la versión Java EE 7 se agregaron múltiples mejoras a las APIs ya existentes y algunas nuevas APIs como las que se ven en color naranja.

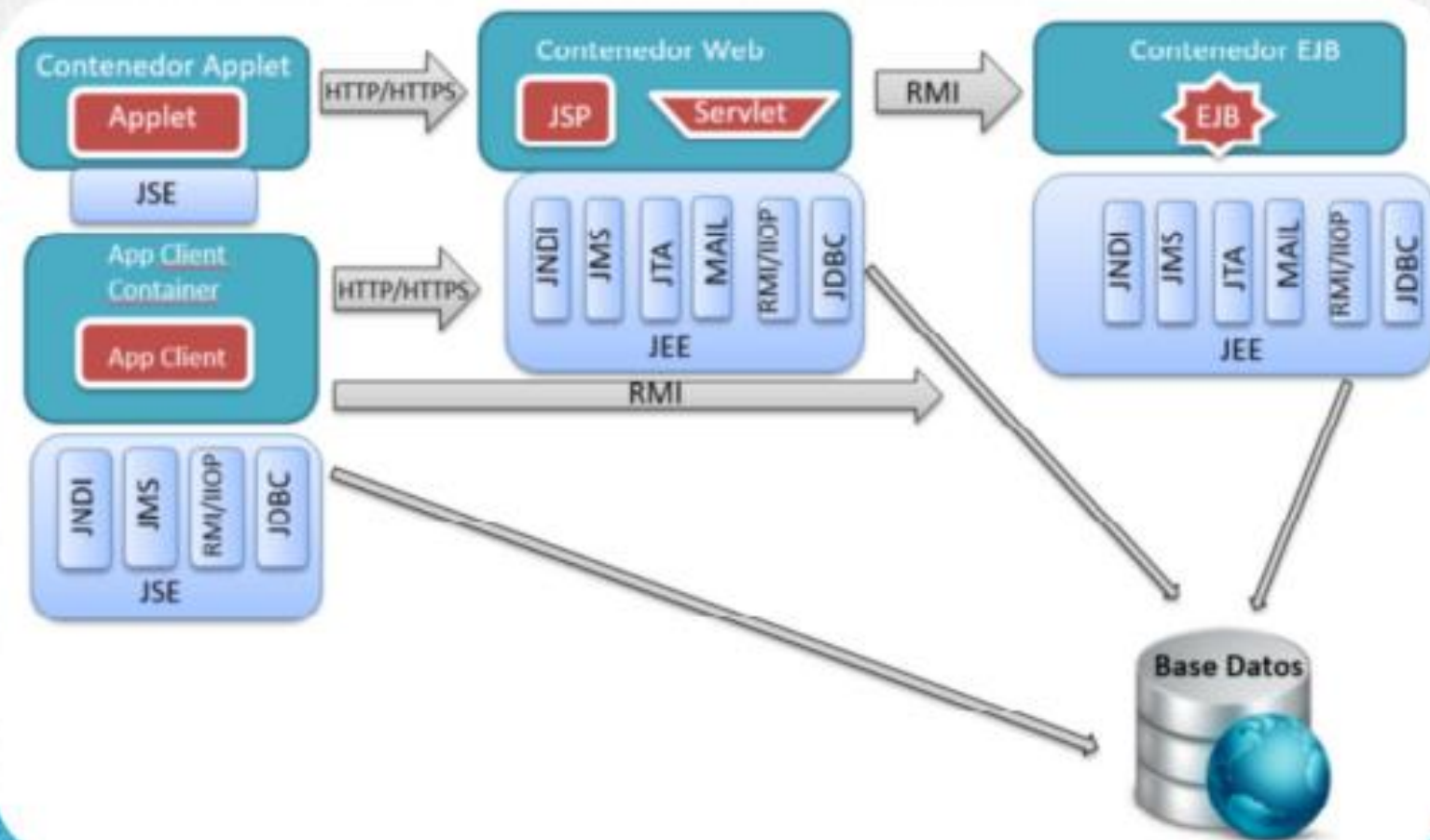
De las mejoras más sustanciales es el API de Web Socket, la cual permite una mejor comunicación entre el cliente y el servidor a través del protocolo HTTP. Aplicaciones de tipo chat o mensajería sobre una aplicación Web es ahora más fácil de implementar gracias a esta API, especialmente impactando en la mejora de aplicaciones en tiempo real.

También una de las mejoras más importantes es poder ejecutar las aplicaciones empresariales en la nube ya sea de manera privada o pública, ya que este es uno de los escenarios donde Oracle ha puesto mayor énfasis.

También se agregó una mejor integración con tecnologías como HTML5, WebSockets, JSON, entre varios estándares más.

También se removieron varias inconsistencias entre las versiones anteriores y la nueva versión, para que los componentes de tipo Managed Beans, EJB, Servlets, JSF, CDI, etc, puedan convivir sin problemas en esta versión.

# TECNOLOGÍAS EMPRESARIALES JAVA EE



La Tecnología Empresarial JEE 7 incluye muchas mejoras en cada una de las tecnologías que la componen, en particular se enfoca en simplificar la integración de varios componentes.

Algunas de las tecnologías más importantes son:

- Enterprise JavaBeans 3.2 (EJB)
- Servlets 3.1
- JavaServer Pages 2.3
- JavaServer Faces (JSF) 2.2
- Expression Language 3.0
- Java Persistence API 2.1
- Java Transaction API (JTA) 1.2
- Java API from XML Web Services (JAX-WS) 2.2
- Java API for RESTful Web Services (JAX-RS) 2.0
- Context and Dependency Injection (CDI) 1.1
- Java Message Service API (JMS) 2.0
- JavaMail API 1.5
- Entre varias tecnologías más.

# JAVA ENTERPRISE EDITION

En una aplicación JEE podemos utilizar tecnologías como EJB, JPA, Web Services, entre muchas más.



En una aplicación JEE podemos utilizar tecnologías como EJB, JPA, Web Services, entre muchas más. En una aplicación empresarial Java podemos manejar características tales como:

- Empaquetar EJB locales en un archivo .war
- Singleton Session Beans
- EJB lite, que son EJB simplificados para utilizar en aplicaciones Web
- Integración entre JSF y EJB a través de CDI

Otras características en la versión Java EE son:

- **Contenedor Ligero:** Para hacer más ligero el contenedor se introdujo el concepto de perfiles, donde podemos seleccionar el conjunto mínimo de tecnologías a utilizar. Por ejemplo si una aplicación Web necesita de EJB, pero no de Servicios Web se puede utilizar EJB Lite, el cual se enfoca en utilizar únicamente las características básicas de los EJB, las cuales estudiaremos en lecciones posteriores.
- **Remover APIs antiguas (Pruning):** Se planteó que para versiones futuras de Java EE, se eliminen algunas API's, ya que son tecnologías mucho más complejas que las nuevas. Por ejemplo, EJB 2.x Entity Beans, ha sido sustituidos por JPA. El API de JAX-RCP se sustituyó por JAX-WS, y así varias tecnologías han sido reemplazadas. Por ello puede ser que en futuras versiones ya no sean soportadas.

- **Facilidad de Uso:** El uso de anotaciones simplificó en gran medida el uso de configuración vía archivos xml, por lo que archivos como faces-config.xml, ejb-jar.xml y persistence.xml se redujeron a unas cuantas líneas, e incluso en algunos casos es opcional el utilizarlos. Además las clases ahora están orientadas a clases puras de Java (POJOs) e Interfaces, y en algunos casos, como los EJB's, el uso de interfaces es opcional. Sin embargo, al igual que en Spring Framework, el uso de interfaces es una buena práctica que se sigue aplicando al día de hoy.
-

# ARQUITECTURA MULTICAPAS EN JAVA EE





Una aplicación empresarial en Java se compone de distintas capas, cada capa tiene una función muy específica. Dividir una aplicación en capas tiene varias ventajas, como son separación de responsabilidades, un mejor mantenimiento a la aplicación, especialización de los programadores en cada capa, entre muchas más.

---

La versión empresarial de Java brinda un API distinta para cada capa de una aplicación empresarial, desde la capa de presentación, la capa de negocio y la capa de datos.

A continuación mencionaremos cada una de las capas de una aplicación multicapas.

- **Capa Web:** La capa del Cliente es donde el cliente interactúa por medio de un navegador Web, un cliente móvil, una aplicación de escritorio, entre otros.
- **Capa Web:** la capa web que puede residir en un servidor web, las tecnologías mas básicas que podemos encontrar en este servidor web son los JSP's y los Servlets o JavaServer Faces.
- **Capa de Negocio:** en esta capa podemos encontrar tecnología como son los Enterprise Java Beans (EJBs).
- **Capa de Datos:** Aquí vamos a encontrar tecnologías como JDBC, o JPA. Este código nos va a permitir comunicarnos con nuestra base de datos para leer y almacenar información en ella.