

Java empresarial y las bases de datos

Java empresarial y las bases de datos	1
¿Qué aprenderás?	2
Introducción	2
Conceptos básicos sobre bases de datos	3
Instalación base de datos Oracle 18c express Edition	4
Instalación en Windows	4
Creación de base de datos en Oracle	8
Instalación en Linux (CentOS, Oracle Linux)	14
Instalación en Mac (Docker, VM)	14
¿Qué es Docker?	14
DBEAVER	15
Conociendo el IDE para conectar a la Base de datos	16
SQL Developer	19
Descargar la opción Windows 32-bit/64-bit	19
Conociendo el IDE para conectar a la Base de datos	22
Conectar programas java con base de datos con JDBC	26
Componentes del api JDBC	27
ResultSet	27
PreparedStatement	28
Connection	29
Drivers de conexion con Oracle	30
Oracle JDBC Thin Driver	30
Oracle JDBC OCI Driver	30
Descarga del driver Oracle JDBC	31



¡Comencemos!

¿Qué aprenderás?

- Conocer los conceptos básicos sobre base de datos.
- Instalar el motor Oracle 18c.
- Crear una base de datos en Oracle.
- Conocer la IDE de BD Dbeaver.
- Conectar aplicaciones Java con base de datos.

Introducción

Toda aplicación debe interactuar con una base de datos, y es obligatorio para un desarrollador conocer las distintas técnicas utilizadas para llevar a cabo este cometido. Entrar en el mundo de la capa de persistencia en un sistema java involucra no solo conocer sobre sentencias SQL o conocer las capas de forma teórica, también involucra familiarizarse con los entornos de trabajo con bases de datos, con las herramientas que permiten administrar las tablas, con las formas de comunicación entre aplicación y motores de bases de datos, con el concepto de driver de conexión y en general con todo el ecosistema que si o si uno se encuentra en un equipo de desarrollo.

Es importante conocer cuáles son las herramientas que los desarrolladores java full stack utilizan en su día a día y entender su funcionamiento desde la base, para así, con práctica, adquirirlas y añadirlas al arsenal de capacidades que un profesional debe tener. Se dará a conocer el ambiente de desarrollo y sus herramientas más usadas por el simple hecho de que es muy probable que se encuentren con ellas en la vida real. También se dará un vistazo a los conceptos básicos de las bases de datos para finalizar con la conexión básica que permite que tu aplicación java trabaje junto a una base de datos.

Conceptos básicos sobre bases de datos

Una base de datos es un conjunto de datos almacenados, organizados y ordenados sistemáticamente, que en el mundo real pueden representar información o características de un contexto específico, de una persona, cosa u evento. Hoy en día, las bases de datos están presentes prácticamente en casi todo lo que vemos y/o utilizamos por ejemplo, en redes sociales, aplicaciones móviles, sistemas de acceso, etc.

Podemos comparar una base de datos con una carpeta que contiene las fichas de todos los pacientes de un hospital, ya que cada ficha tendrá el nombre del paciente, edad, dirección, patologías, exámenes, etc. La cual debe estar organizada y ordenada de forma ascendente en base al apellido del paciente.

En informática una base de datos relacional está organizada en tablas, columnas y relaciones, de acuerdo al ejemplo anterior podemos decir que la ficha es una tabla y los datos del paciente son las columnas.

Instalación base de datos Oracle 18c express Edition

Para seguir los ejercicios, en esta parte del curso utilizaremos una base de datos robusta y quizás un poco intimidante como es Oracle en su versión 18c bajo Windows. Si bien una buena instalación cuesta un poco, al final del día es lo mejor, ya que nos enfrentamos a configuración real y además, conoceremos dos buenos IDEs de base de datos, el **DBEAVER** y el **SQL DEVELOPER**, los cuales son muy utilizados por los profesionales en la industria.

Instalación en Windows

Para la instalación, hay que tener unas consideraciones:

- No debe existir ningún resto de instalación de alguna versión de Oracle, ya que si es así el instalador no podrá finalizar. Debe estar instalado el JDK de Java.

Nos dirigimos a la página de descarga de Oracle 18c y descargamos la versión indicada.

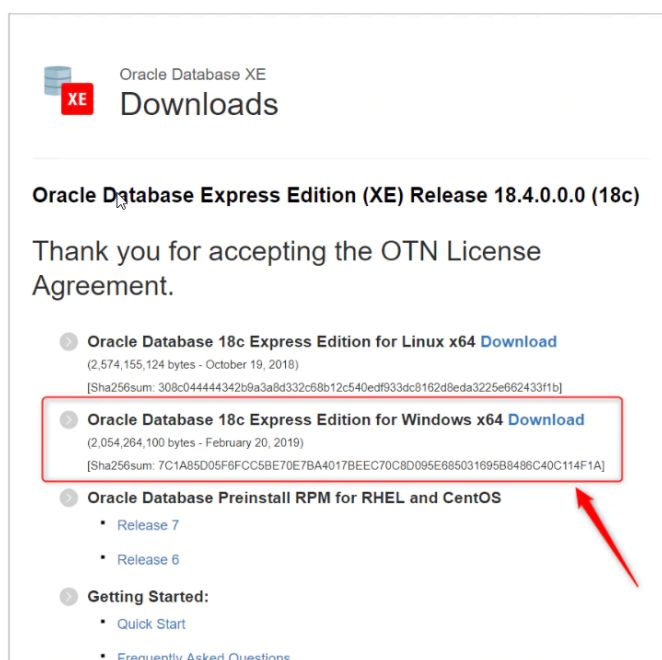


Imagen 1. Instalando Oracle Database.

Fuente: Desafío Latam.

Descomprimos el archivo obtenido y lo guardamos en alguna ubicación conocida. Dentro de esos archivos busca un ejecutable de nombre setup.exe. Lo ejecutas como administrador y verás el asistente de instalación:

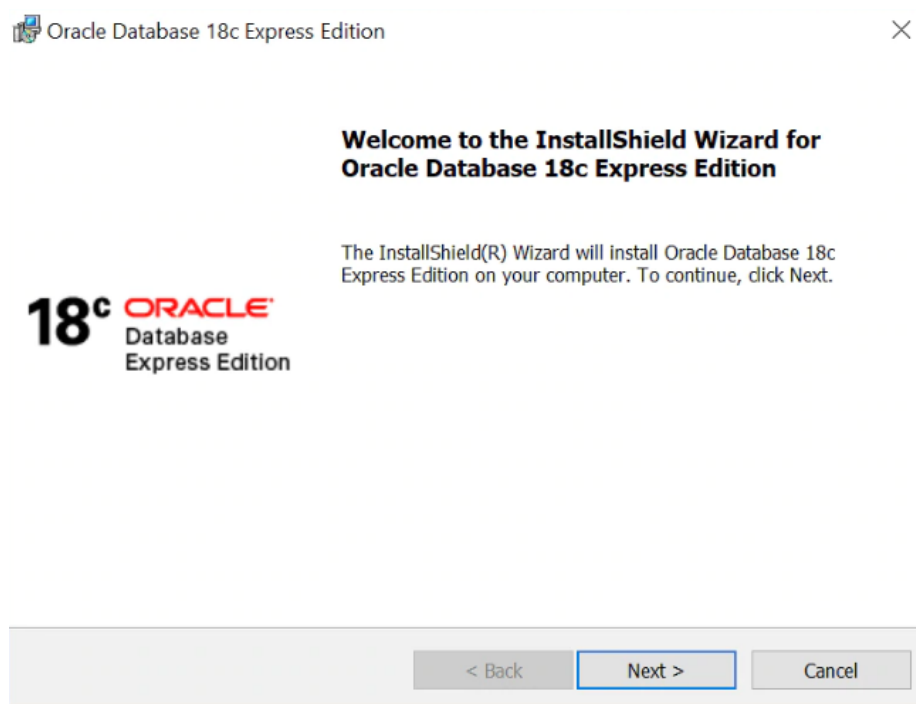


Imagen 2. Instalador.
Fuente: Desafío Latam.

Al seleccionar next, el asistente hará las comprobaciones de rigor del ambiente y por ejemplo, si encuentra alguna otra instalación de oracle detendrá el asistente. Si está todo bien, mostrará la siguiente ventana en donde debemos indicar la ubicación de la futura instalación.

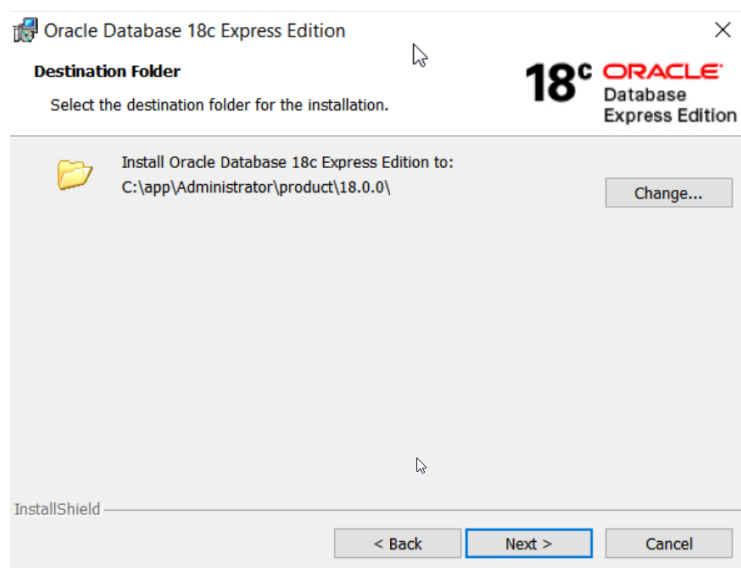


Imagen 3. Elijiendo directorio.

Fuente: Desafío Latam.

En la siguiente ventana del asistente, se pregunta el password para los usuarios por defecto de la base de datos. Ten en cuenta que en las instalaciones Oracle los usuarios por defecto son:

- SYS.
- SYSTEM o PDBADMIN (PDB es la sigla de pluggable database).
Para este caso deja el password como admin (es fácil de recordar, ya que un grave error es instalar oracle y olvidar la clave y el usuario, que siempre será SYSTEM para conectar, ojo con eso).

Oracle recomienda que la contraseña ingresada debe tener:

- Al menos 12 caracteres de longitud.
- Contener al menos 1 carácter en mayúscula.
- 1 carácter en minúscula.
- 1 dígito [0-9].

Por ser pruebas didácticas utilizamos una clave poco segura.

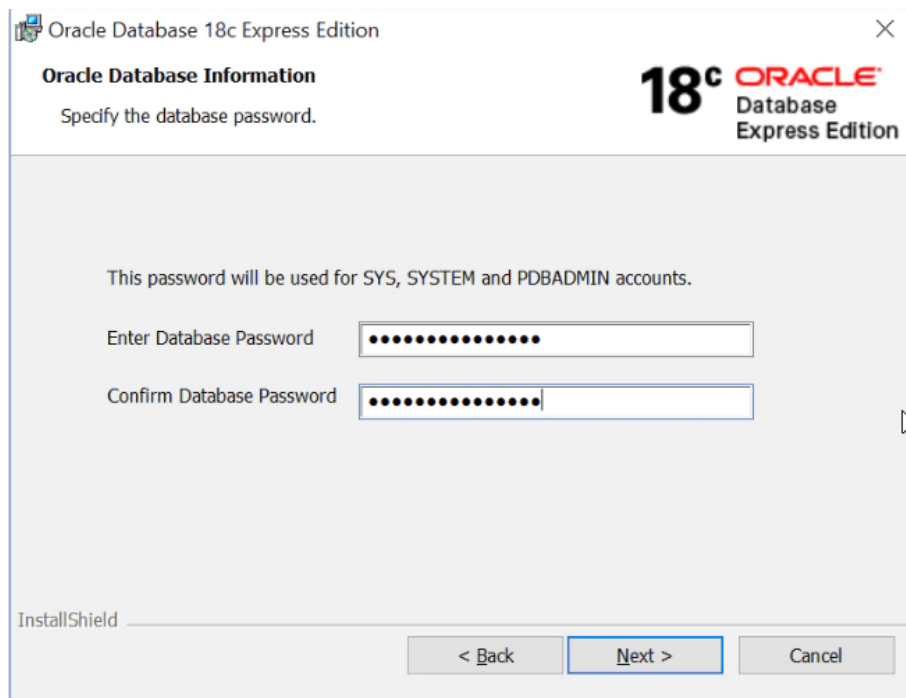


Imagen 4. Agregando contraseña a la base de datos.
Fuente: Desafío Latam.

Luego de configurar el password y pinchar el botón install, oracle se instalará en la máquina. Esperamos unos minutos y al final mostrará una pantalla de confirmación con el resumen de la instalación.

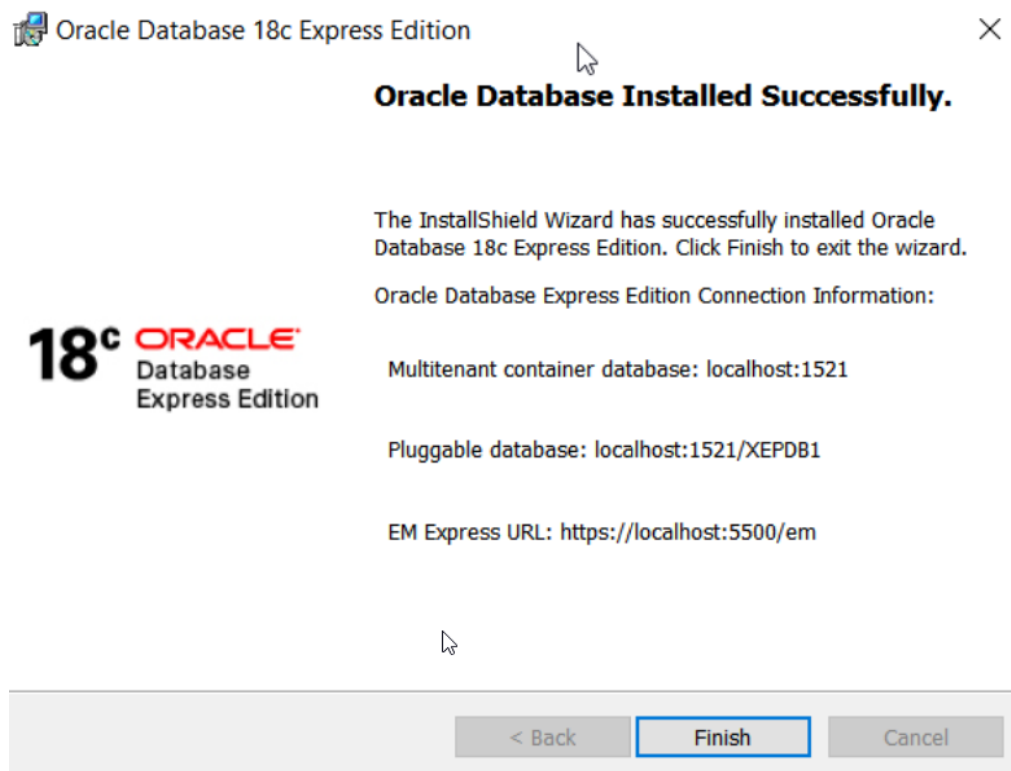


Imagen 5. Resultado de la instalación.

Fuente: Desafío Latam.

Ya está el motor Oracle en el sistema. Ahora procedemos a instalar el programa que nos permitirá administrar a Oracle.

Creación de base de datos en Oracle

En Oracle, para crear una base de datos se utiliza un asistente de creación de base de datos conectable (Pluggable Database) el cual da las instrucciones para configurarla correctamente, ya que no es tan sencillo como en mysql u otros motores dada su naturaleza empresarial. El asistente fue instalado en la máquina junto a la instalación anterior por lo cual para acceder a ella debes ir al botón inicio de windows y ubicar la instalación de Oracle y seleccionar Database Configuration Assistant.

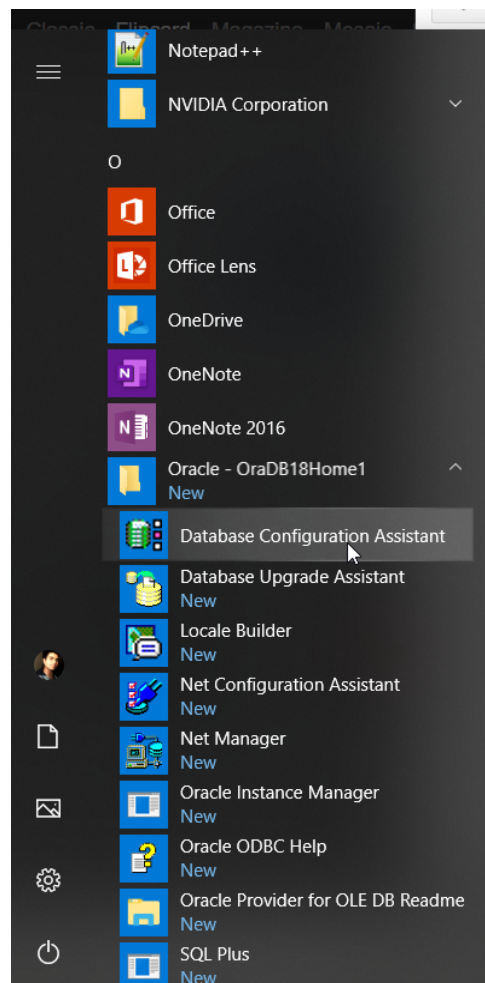


Imagen 6. Creando una base de datos.
Fuente: Desafío Latam.

Se desplegará el asistente de creación y administración de base de datos.

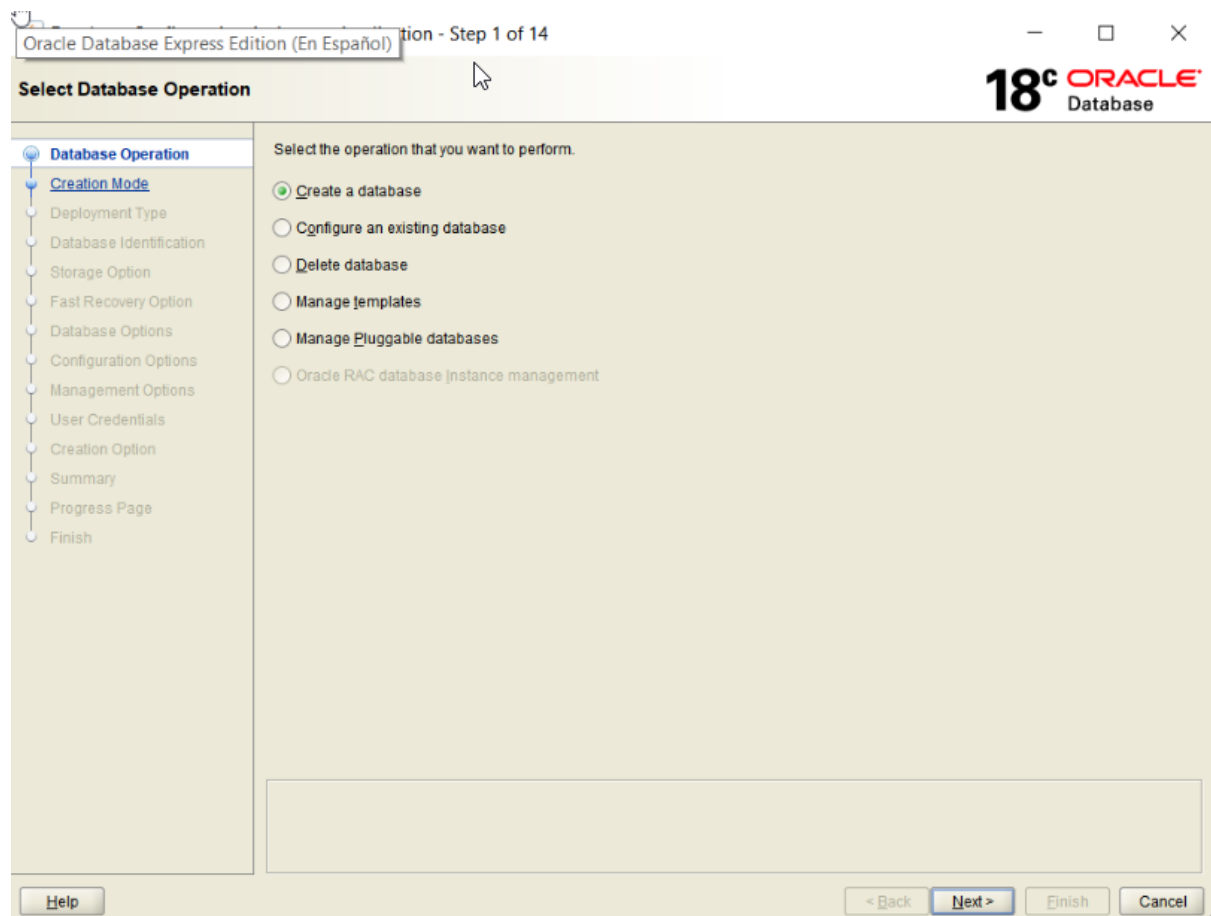


Imagen 7. Asistente de creación.
Fuente: Desafío Latam.

Deberás seleccionar la opción Manage Pluggable database.

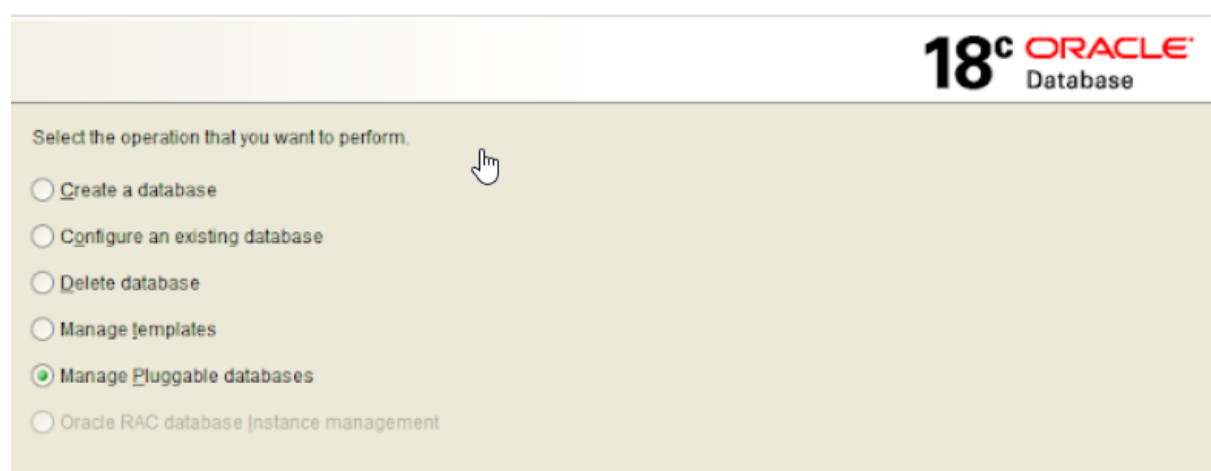


Imagen 8. Seleccionar Manage Pluggable database.
Fuente: Desafío Latam.

Luego, selecciona Create a pluggable database.

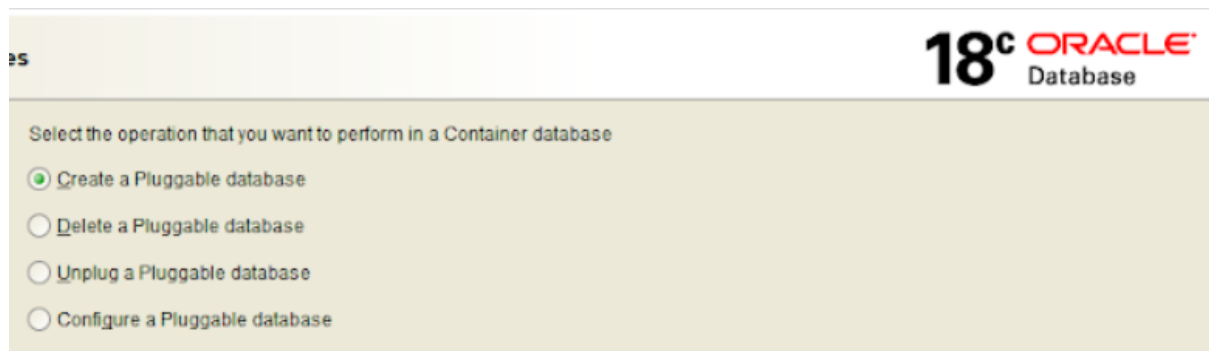


Imagen 9. Seleccionar Create a Pluggable database.

Fuente: Desafío Latam.

Ingresa la contraseña del usuario sys con la instancia XE seleccionada.

18^c ORACLE[®] Database

Select a Container database within which Pluggable database needs to be created.

	Database	Local instance	Type
<input checked="" type="radio"/>	XE	XE	Single instance

DBCA will connect to the database using OS based authentication. Database credentials may be needed if OS based authentication is disabled. Specify the credentials, if needed.

User name:

Password:

< Back Next > Finish Cancel

Imagen 10. Seleccionando la base de datos XE.

Fuente: Desafío Latam.

Selecciona Create a new pluggable database from another y PDB\$SEED en el combo de selección.

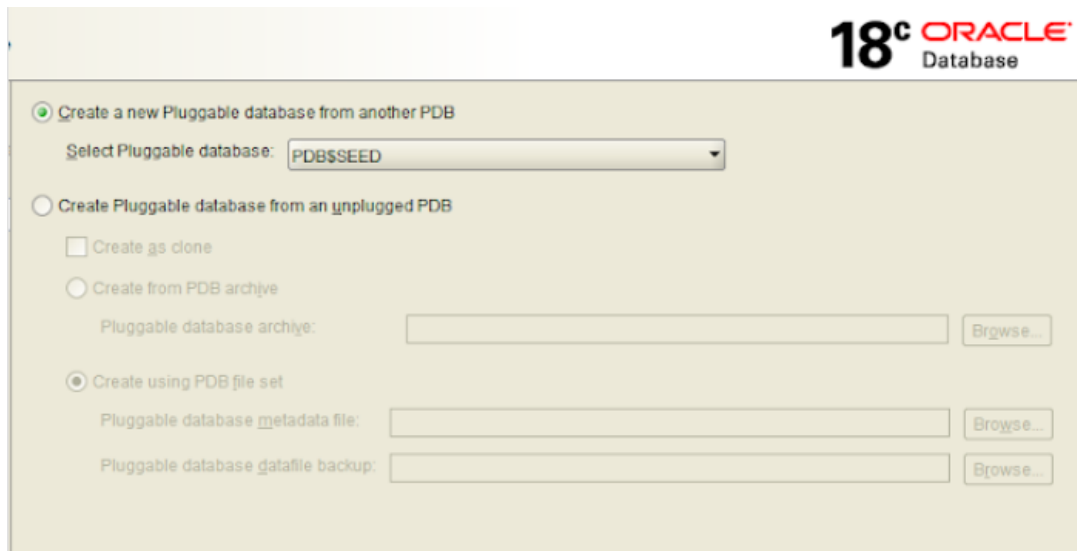


Imagen 11. Seleccionar PDB\$SEED.
Fuente: Desafío Latam.

Da un nombre a la nueva base de datos (PDB) y un nombre de usuario con su respectiva contraseña:

- **Nombre de base de datos:** desafio_ejemplo01.
- **UserName administrador:** desafio.
- **Password:** desafio_latam.

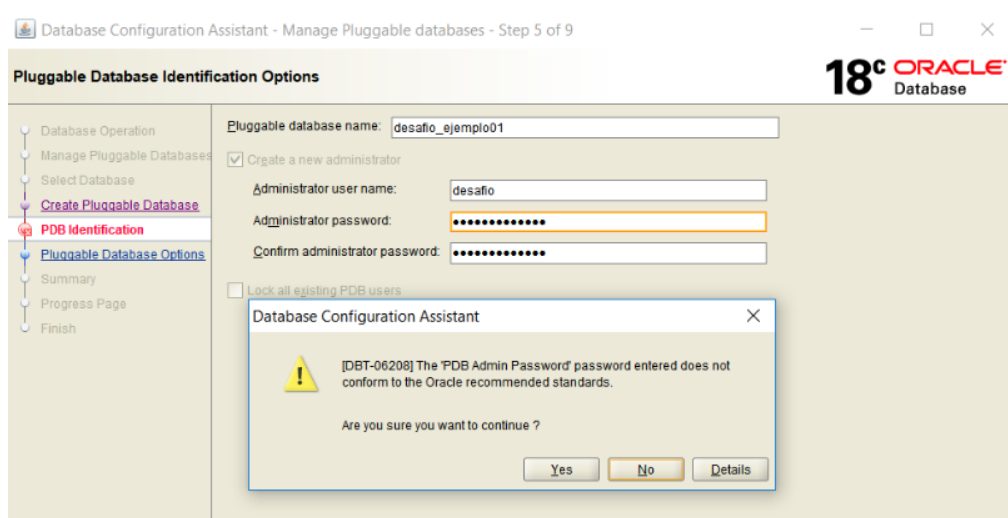
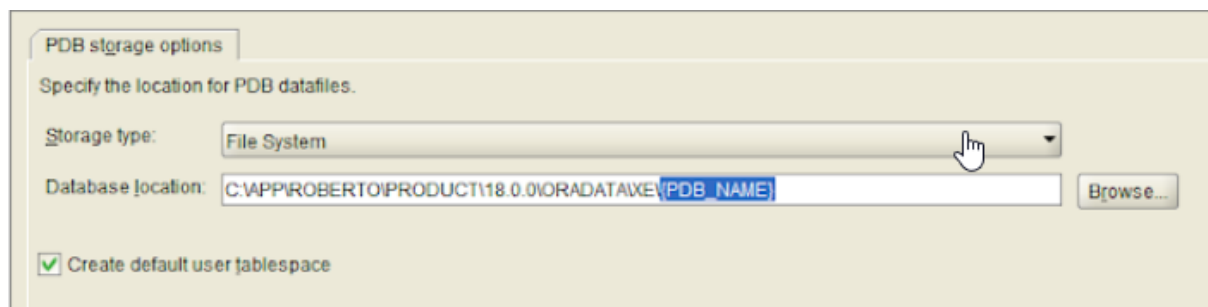


Imagen 12. Indica que no cumple con los requisitos.
Fuente: Desafío Latam.

El asistente indicará que la clave no cumple con los requisitos, sin embargo, vamos continuar de todas formas.

En esta opción, reemplaza el área que dice PDB_NAME por el nombre de nuestra base de datos. Selecciona también el check que indica la creación del tablespace de usuario. Finalizamos y esperamos a que el asistente termine las configuraciones. Al final tendrás una nueva base de datos bajo el control del usuario previamente creado.



PDB storage options

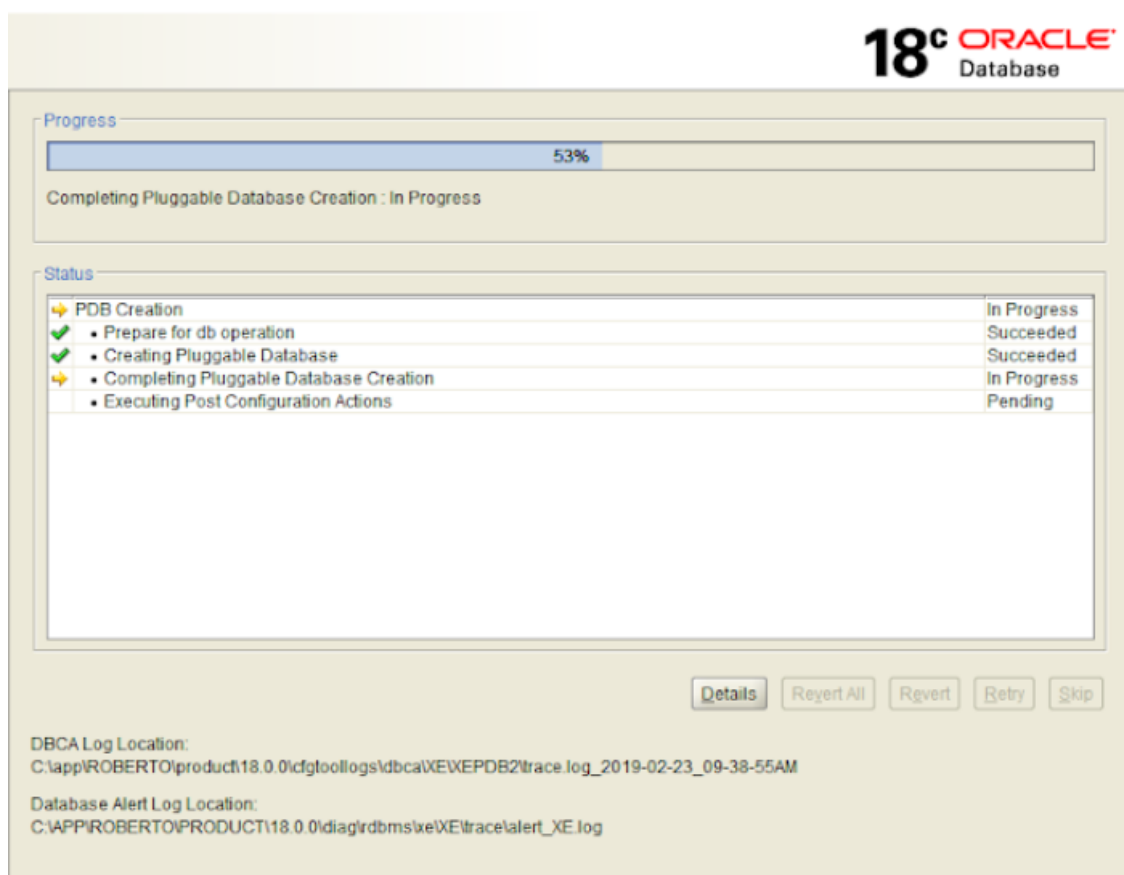
Specify the location for PDB datafiles.

Storage type: File System

Database location: C:\APP\ROBERTO\PRODUCT\18.0.0\ORADATA\XE\{PDB_NAME} Browse...

☒ Create default user tablespace

Imagen 13. Agregar nombre a la base de datos.
Fuente: Desafío Latam.



18c ORACLE Database

Progress

53%

Completing Pluggable Database Creation : In Progress

Status

➔ PDB Creation	In Progress
• Prepare for db operation	Succeeded
• Creating Pluggable Database	Succeeded
• Completing Pluggable Database Creation	In Progress
• Executing Post Configuration Actions	Pending

Details Revert All Rgvert Retry Skip

DBCA Log Location:
C:\app\ROBERTO\product\18.0.0\cfgtoollogs\dbca\XE\XPDB2\trace.log_2019-02-23_09-38-55AM

Database Alert Log Location:
C:\app\ROBERTO\PRODUCT\18.0.0\diag\rdms\xe\XE\trace\alert_XE.log

Imagen 14. Creando la base de datos.
Fuente: Desafío Latam.

Perfecto, ahora es tiempo de conocer nuestra herramienta de base de datos DBEAVER o más conocido en el ambiente como “la rata” (aunque es un castor).

Instalación en Linux (CentOS, Oracle Linux)

CentOS 8:

- <https://tipstecnologicos.es/linux-instalacion-oracle-18c-xe-en-centos-8-mediante-con-sola/>

Oracle Linux:

- <http://oracleparatodos1.blogspot.com/2018/05/instala-oracle-18c-sobre-linux-6-y-cr-ea.html>
- <https://oraxedatabase.blogspot.com/2018/10/como-instalar-oracle-database-18c.html>
- <https://apasoft-training.es/blog/instalar-oracle-18c-en-linux>

Instalación en Mac (Docker, VM)

¿Qué es Docker?

Docker es un contenedor virtual diseñado para facilitar la creación, implementación y ejecución de aplicaciones mediante el uso de contenedores. Los contenedores permiten a un desarrollador empaquetar una aplicación con todas las partes que necesita, como bibliotecas y otras dependencias, y desplegarla como un paquete.

Al hacerlo, gracias al contenedor, el desarrollador puede estar seguro de que la aplicación se ejecutará en cualquier otra máquina o servidor. Las principales características de un contenedor Docker son: no posee sistema operativo como una máquina virtual, puede definir sus propias variables de entorno, puede configurar su propia infraestructura de red y puede configurar su propia estructura de directorios o carpetas.

Referencias a instalaciones con Docker:

- <https://medium.com/@mfofana/how-to-install-oracle-database-on-mac-os-sierra-10-12-or-above-c0b350fd2f2c>
- <https://danthesalmon.com/running-oracle-12c-on-macos/>
- <https://www.databasesstar.com/oracle-mac/>

DBEAVER

Descripción de la herramienta:

- Es una herramienta de administración de base de datos libre y basada en java.
- Tiene muchas funcionalidades y un muy buen funcionamiento.
- Puede utilizarse no solo con Oracle sino que también con mysql, postgres, bases de datos no relacionales, etc.
- Es una herramienta muy usada en la industria por los desarrolladores dada su facilidad de uso y su bajo consumo de recursos.

Para descargarlo dirígite a: <https://dbeaver.io/download/> y ubica el archivo en alguna ubicación conocida.

No necesita instalación por lo que solo debes seleccionar el icono de dbeaver (un castor y se iniciará el programa).



Imagen 15. DBEAVER.

Fuente: Desafío Latam.

Conociendo el IDE para conectar a la Base de datos

La interfaz es muy similar a eclipse y es muy intuitiva. En este caso nos enfocaremos en el sector izquierdo.

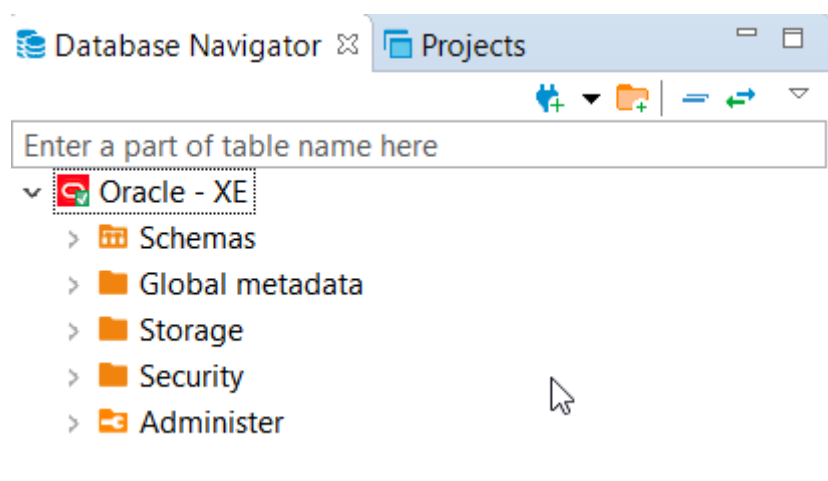


Imagen 16. Interfaz.
Fuente: Desafío Latam.

Observación: DBEAVER al principio puede que pida unos drivers de Oracle si es que no los encuentra en la máquina. El mismo programa te indicará que faltan controladores de oracle y te dará la opción de descargarlos o añadirlos desde alguna ubicación de tu equipo.

Vemos un árbol de directorios y el *tablespace XE*, que está disponible por defecto en la instalación de Oracle. Pero nosotros queremos trabajar con nuestro propio usuario oracle y la base de datos que se creó en pasos posteriores. Para hacer eso:

- Ubicar la opción *New DataBase Connection*.
- Se desplegará un cuadro de nueva conexión para que elijamos nuestro motor de destino.
- Seleccionar Oracle y continuar.

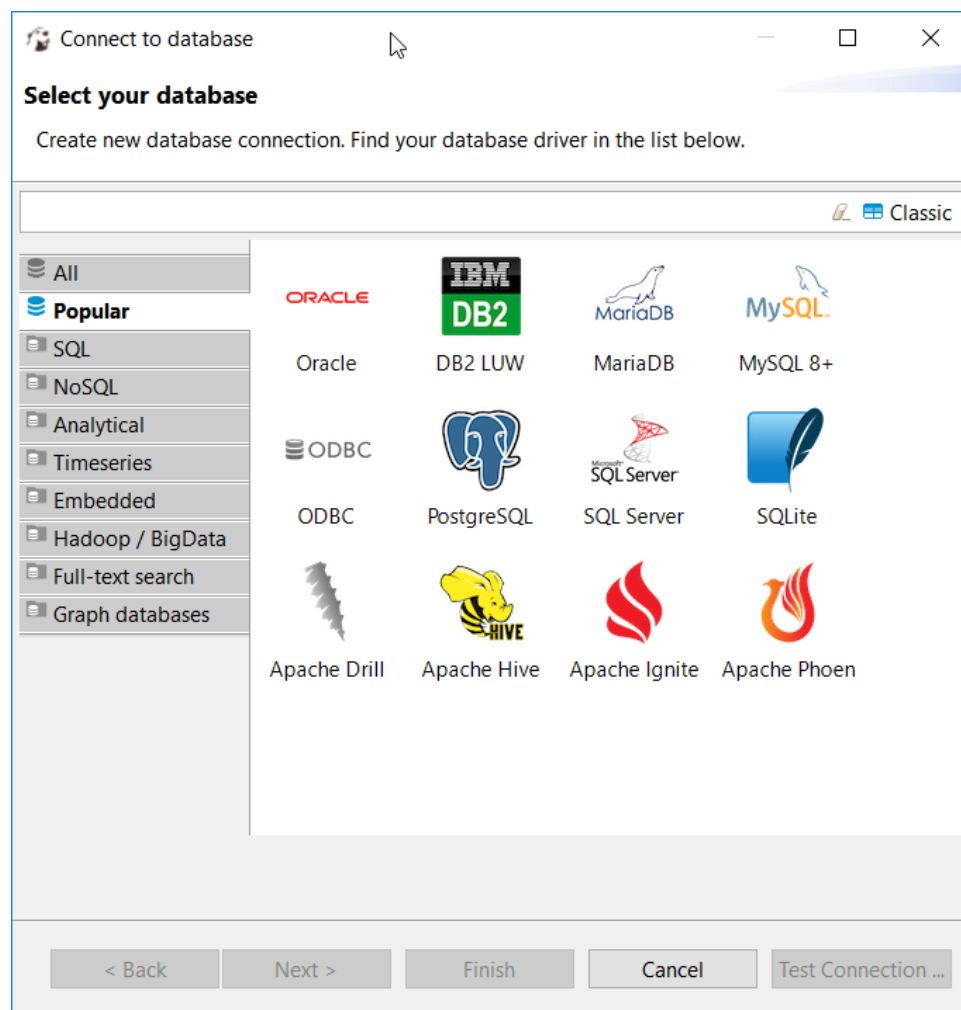
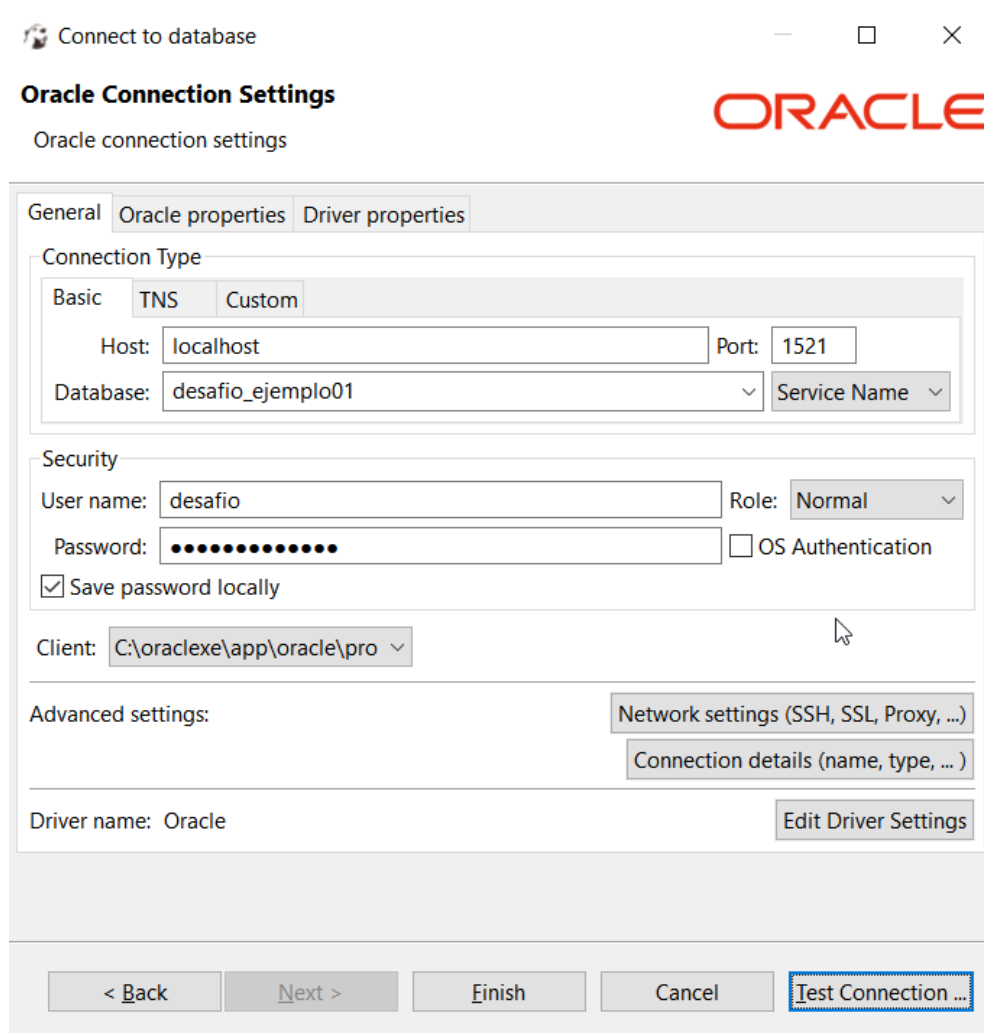


Imagen 17. Seleccionar motor de base de datos a utilizar.

Fuente: Desafío Latam.

- Se desplegará un cuadro de configuración de conexión.
- Usamos las credenciales configuradas en el paso de creación de PDB (*Pluggable Data Base*).
- En la opción *Client* seleccionar la ubicación de nuestra instalación Oracle.



The image shows the 'Connect to database' dialog box with the 'Oracle Connection Settings' tab selected. The 'General' sub-tab is active, showing 'Connection Type' set to 'TNS'. The 'Host' is 'localhost', 'Port' is '1521', and 'Database' is 'desafio_ejemplo01'. The 'Security' section shows 'User name' as 'desafio', 'Role' as 'Normal', and 'OS Authentication' is unchecked. The 'Client' path is 'C:\oracle\app\oracle\pro'. At the bottom, the 'Test Connection ...' button is highlighted with a red border.

Connect to database

Oracle Connection Settings

Oracle connection settings

General Oracle properties Driver properties

Connection Type

Basic TNS Custom

Host: localhost Port: 1521

Database: desafio_ejemplo01 Service Name

Security

User name: desafio Role: Normal

Password: OS Authentication

☒ Save password locally

Client: C:\oracle\app\oracle\pro

Advanced settings: Network settings (SSH, SSL, Proxy, ...) Connection details (name, type, ...)

Driver name: Oracle Edit Driver Settings

< Back Next > Finish Cancel Test Connection ...

Imagen 18. Probando la conexión.
Fuente: Desafío Latam.

El botón Test Connection indicará si los datos son correctos.

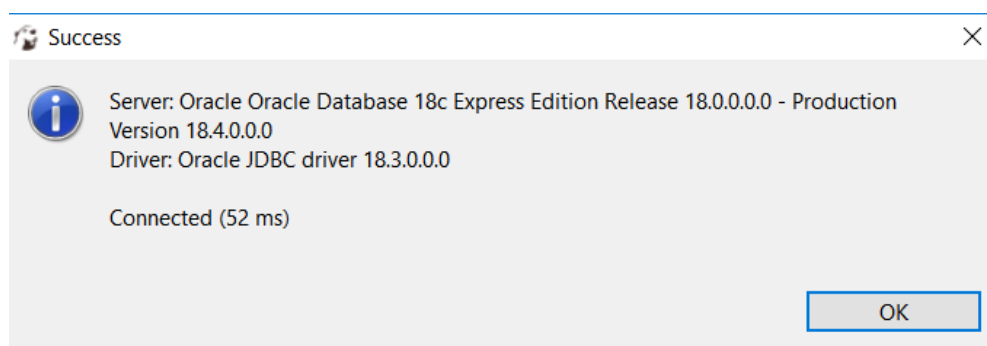


Imagen 19. Resultado al conectarse a la base de datos.
Fuente: Desafío Latam.

Ya tenemos nuestra base de datos con sus esquemas bajo el usuario que definimos. Ahora podemos cómodamente utilizar *dbeaver* para administrar la base de datos. Si desean crear más bases de datos solo sigan estos pasos.

SQL Developer

Descripción de la herramienta:

- Es la segunda herramienta más usada en el área es el sql developer, de propiedad de los señores de Oracle.
- Es un entorno de desarrollo integrado y gratuito que simplifica el desarrollo y la gestión de Oracle Database en implementaciones tradicionales y en la nube.
- Ofrece un completo desarrollo integral para sus aplicaciones PL/SQL, una hoja de trabajo para ejecutar consultas y scripts, una consola DBA para gestionar la base de datos, una interfaz de informe, una solución completa de modelado de datos y una plataforma de migración para mover sus bases de datos de terceros a Oracle.

Para instalar este ide primero debemos descargarlo de la página oficial de oracle:
<https://www.oracle.com/tools/downloads/sqldev-v192-downloads.html>

Descargar la opción Windows 32-bit/64-bit

SQL Developer 19.2 Downloads

Version 19.2.0.206.2117 August 1, 2019

- [Release Notes](#)
- [Bugs Fixed](#)
- [Documentation](#)



Platform	Download	Notes
Windows 64-bit with JDK 8 included	 Download (490 MB)	<ul style="list-style-type: none">• MD5: aae291a468b73fc5c374f9213d7456d• SHA1: f250948419d3336998fb666f8ce5dcd6203431ab• Installation Notes
Windows 32-bit/64-bit	 Download (410 MB)	<ul style="list-style-type: none">• MD5: b6633f8488d6c0ce6a345d59a984d63• SHA1: 086c800487af8dcb2599036a3b56b76f8fc0394d• Installation Notes• JDK 8 or 11 required

Imagen 20. Seleccionar la versión de SQL Developer.

Fuente: Desafío Latam.

La descarga es en formato ZIP, por lo cual en una ubicación conocida descomprimirlo.

Como sugerencia dejarlo en la raíz del equipo

Al abrir el contenido de la carpeta veremos la siguiente estructura de carpetas:

Name	Date modified	Type	Size
configuration	8/19/2019 10:07 PM	File folder	
dataminer	8/19/2019 10:07 PM	File folder	
dropins	8/19/2019 10:07 PM	File folder	
dvt	8/19/2019 10:07 PM	File folder	
equinox	8/19/2019 10:07 PM	File folder	
external	8/19/2019 10:07 PM	File folder	
ide	8/19/2019 10:07 PM	File folder	
javavm	8/19/2019 10:07 PM	File folder	
jdbc	8/19/2019 10:07 PM	File folder	
jdev	8/19/2019 10:07 PM	File folder	
jlib	8/19/2019 10:07 PM	File folder	
jviews	8/19/2019 10:07 PM	File folder	
module	8/19/2019 10:07 PM	File folder	
modules	8/19/2019 10:07 PM	File folder	
netbeans	8/19/2019 10:07 PM	File folder	
rdbms	8/19/2019 10:07 PM	File folder	
sleepycat	8/19/2019 10:07 PM	File folder	
sqldeveloper	8/19/2019 10:07 PM	File folder	
sqlj	8/19/2019 10:07 PM	File folder	
svnkit	8/19/2019 10:07 PM	File folder	
icon	7/25/2019 9:17 PM	PNG File	2 KB
sqldeveloper	7/25/2019 9:29 PM	Application	89 KB
sqldeveloper	7/25/2019 9:17 PM	Shell Script	1 KB

Imagen 21. Estructura de carpetas de SQL Developer.

Fuente: Desafío Latam.

Basta con ejecutar el icono *sqldeveloper* para iniciar el programa. Es probable que solicite la ubicación de la instalación del JDK en el equipo. Si es así, otorgar la ruta y continuar.

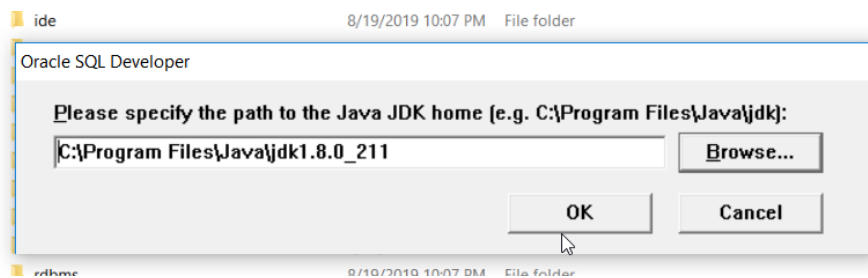


Imagen 22. Seleccionar la ruta del JDK.
Fuente: Desafío Latam.

El programa comienza la carga y al final desplegará el entorno de trabajo completo. Como se pudo ver la instalación es muy fácil e intuitiva.



Imagen 23. Instalando el entorno de trabajo.
Fuente: Desafío Latam.

Conociendo el IDE para conectar a la Base de datos

Ahora solo falta conectar el sql developer con la base de datos correspondiente. Para ello, simplemente dirigirse al sector superior derecho y seleccionar la opción con una cruz verde o también se puede seleccionar con botón derecho en 'oracle connection' y presionar en 'new connection'.

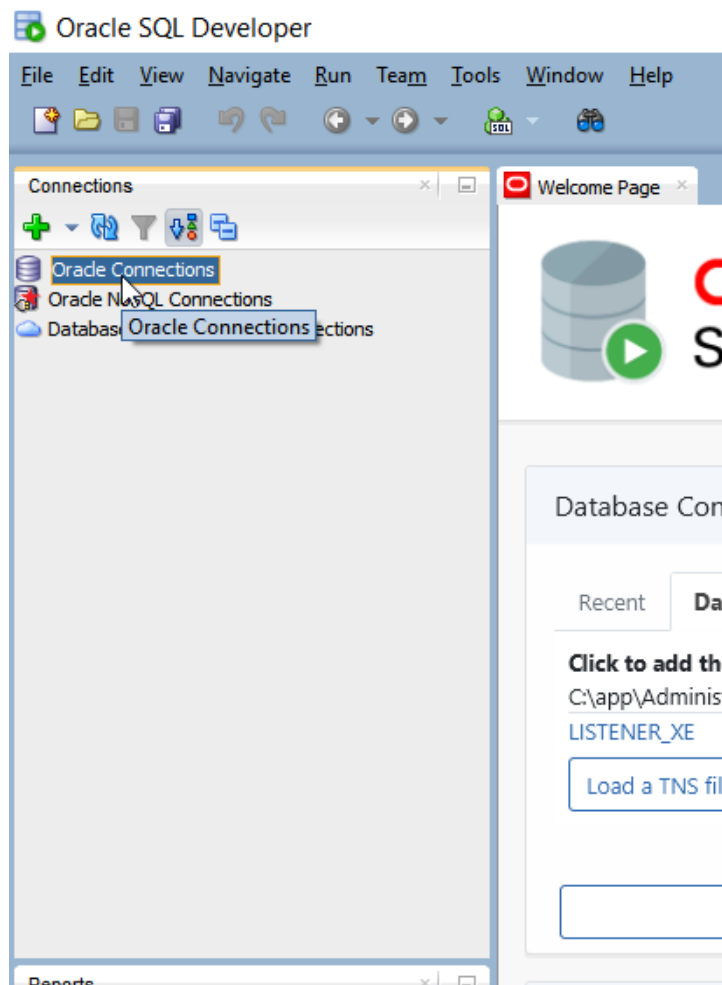


Imagen 24. Creando una conexión.
Fuente: Desafío Latam.

Se mostrará el asistente de conexión en donde espera las credenciales correspondientes.

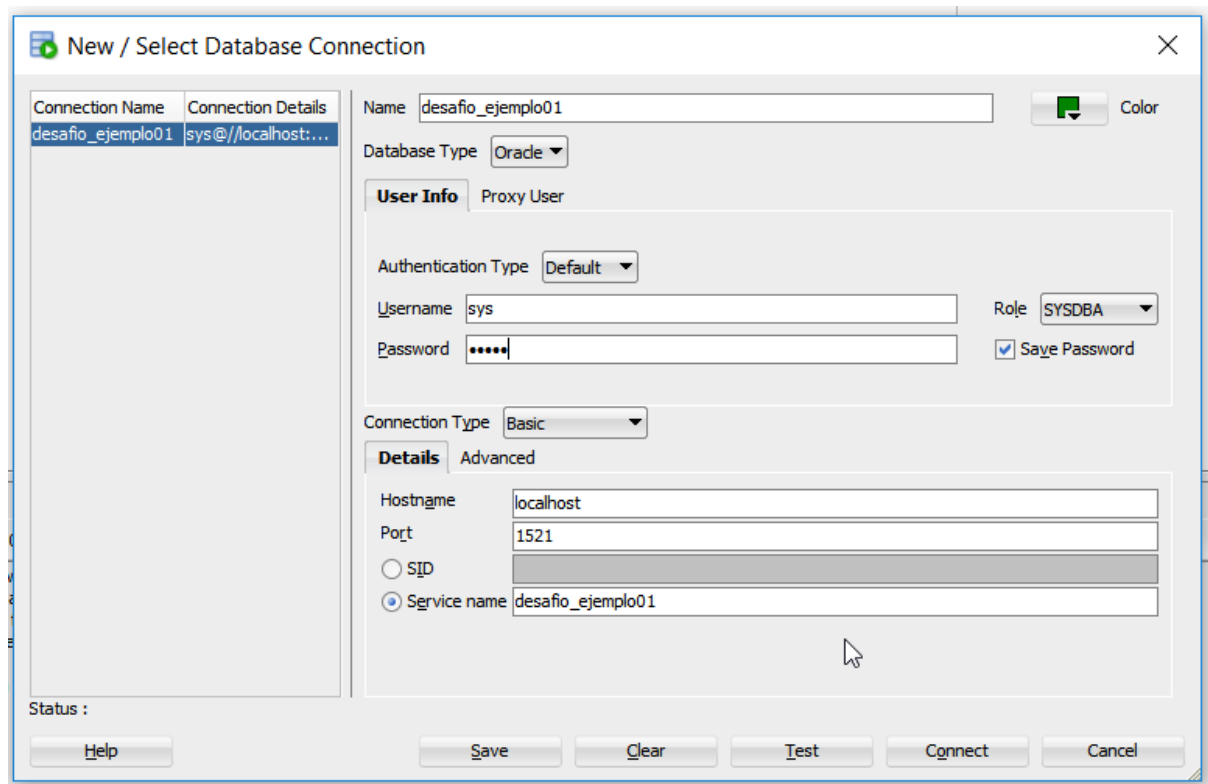


Imagen 25. Agregando las credenciales.
Fuente: Desafío Latam.

Los datos de conexión son:

- **Name:** A gusto personal, puede ser cualquier nombre.
- **Username:** sys.
- **Password:** admin.
- **Role:** SYSDBA.
- **Hostname:** localhost.
- **Port:** 1521.
- **Service name:** desafio_ejemplo01 (es el nombre de la base de datos).

Para probar la conexión, presionar el botón Test y si el resultado es exitoso, presiona en conectar.

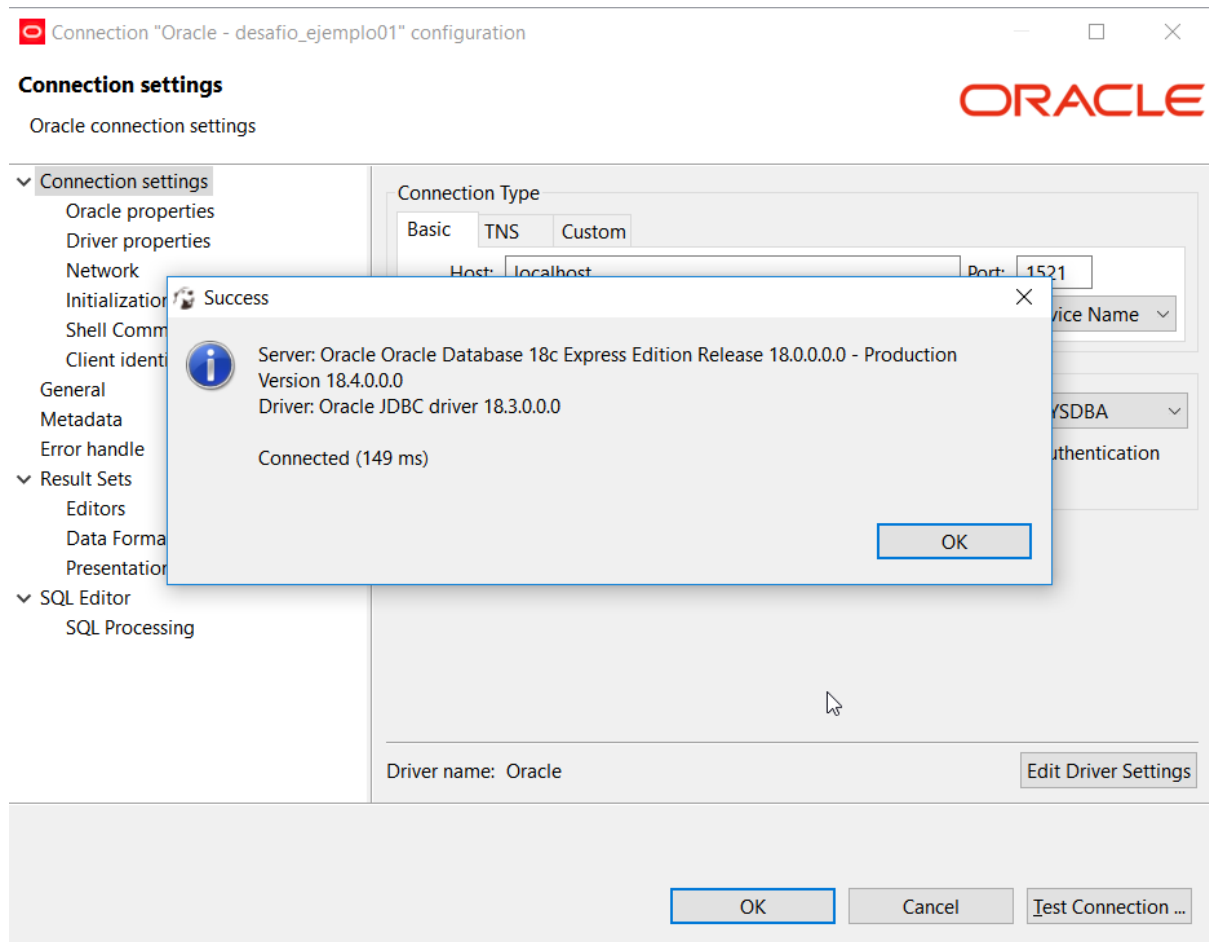


Imagen 26. Probando la conexión.
Fuente: Desafío Latam.

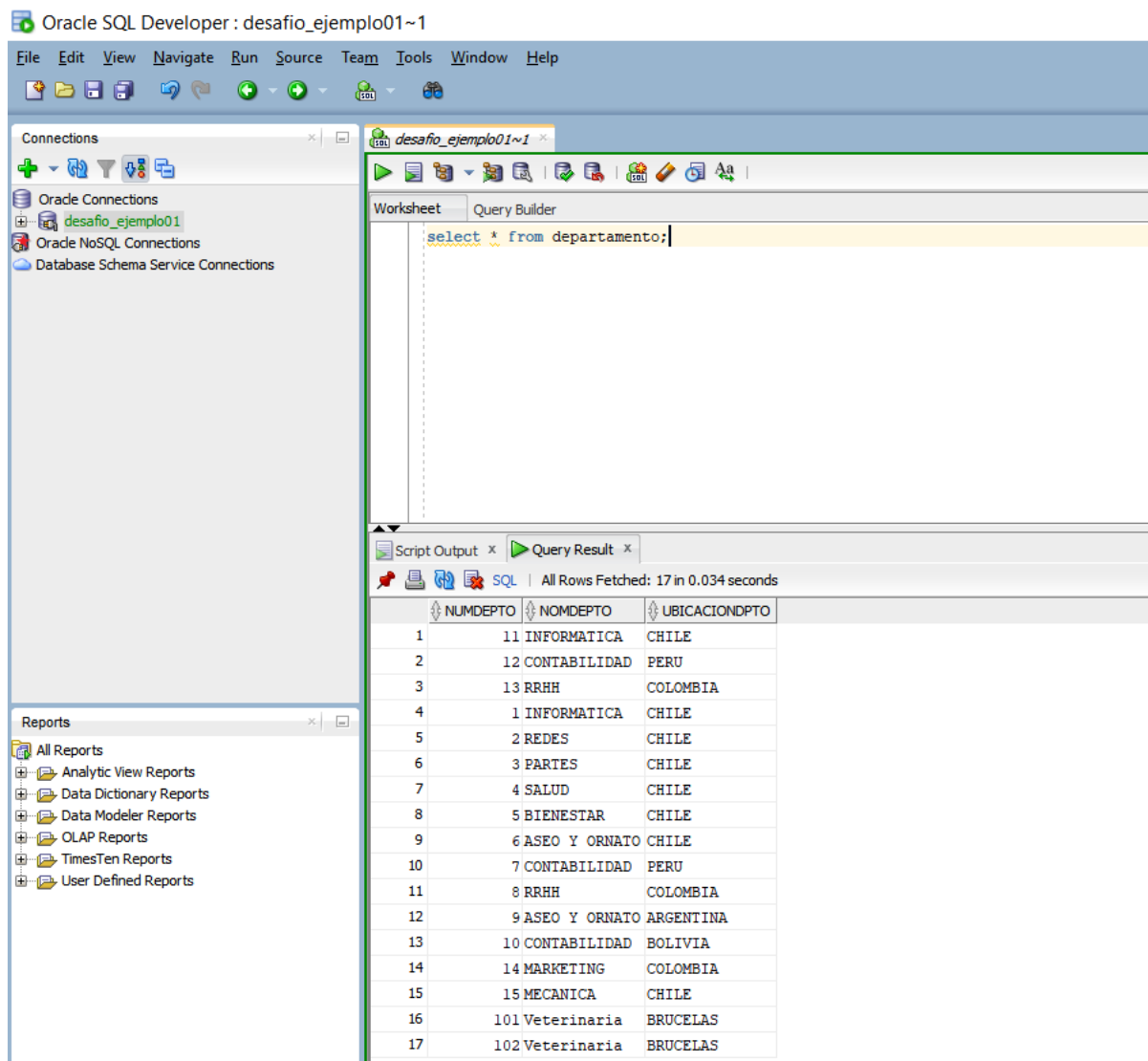


Imagen 27. Realizando una búsqueda.
Fuente: Desafío Latam.

Con estos pasos tenemos las dos aplicaciones instaladas y configuradas. Ahora solo depende de gustos personales.

Conectar programas java con base de datos con JDBC

La API encargada de conectar una aplicación java con un motor de base de datos tiene por nombre JDBC, de data base java connectivity y se encuentra alojado en el paquete java.sql. Dentro de este paquete, encontraremos clases e interfaces a través de las cuales podremos ejecutar queries y updates, invocar procedimientos y funciones y en definitiva, manejar y administrar una base de datos desde el programa java que se esté realizando.

Toda la api JDBC está diseñada sobre interfaces que definen cómo se ejecuta la conexión, la forma de ejecutar las sentencias sql, entre otras funcionalidades. Para poder trabajar junto a una base de datos utilizando esta api, es necesario tener una serie de clases que implementen todas estas interfaces. A este conjunto de clases se le llama drivers.

Un driver de base de datos, para explicarlo de una manera sencilla es un intermediario entre el programa java y el motor de base de datos en cuestión, el cual puede ser Oracle, MySQL, Sql Server etc. Cada fabricante provee sus propios drivers para que un programa usando JDBC pueda interactuar con su motor.

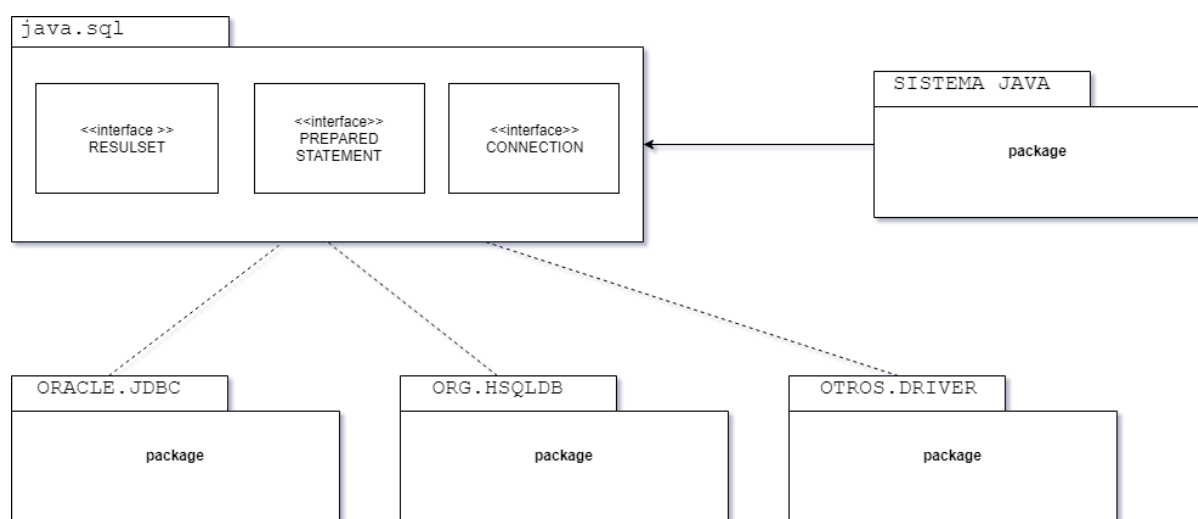


Imagen 28. Diagrama JDBC.

Fuente: Desafío Latam.

Podemos inducir que una aplicación java no está amarrada a solo una base de datos, sino que dependiendo del driver que se implemente puede conectarse a múltiples bases de datos.

Componentes del api JDBC

ResultSet

La interface pública *resultSet* proporciona los métodos necesarios para poder obtener una representación de los datos de una tabla, previa consulta sql. El *resultSet* utiliza un cursor el cual se posiciona después de la primera fila. Con el método *next()* se mueve el cursor a la siguiente fila y cuando no encuentra más filas, devuelve un parámetro *booleano* falso.

Para trabajar con el *resultSet* se utiliza un ciclo *while* en donde se pregunta si el *resultSet* tiene más elementos y mientras la condición se cumpla podemos trabajar con la fila obtenida.

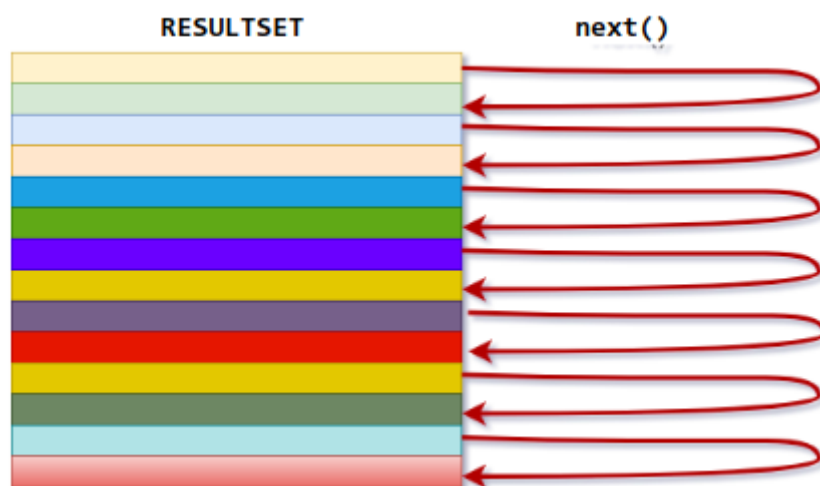


Imagen 29. ResultSet.
Fuente: Desafío Latam.

Un objeto *resultSet* no es actualizable y el cursor solo se mueve hacia adelante. Cuando se recorre esta estructura solo podemos hacerlo desde el primer elemento hasta el último.

Con un *resultSet* podemos recorrer todo un set de datos obtenidos de una tabla gracias a una serie de métodos que proveen distintas funcionalidades. Estos métodos son implementaciones de *setters* y *getters* que nos facilita la recolección de los datos de la tabla.

Por ejemplo, si la *query* retorna un valor de tipo String que se llama nombre, el procedimiento es:

```
resultSet.getString("nombre");
```

Así como existe el método `getString` para las cadenas de texto, también existe un método para cada tipo de dato (`getInt`, `getDouble`, etc.).

int	<code>getConcurrency()</code> Retrieves the concurrency mode of this <code>ResultSet</code> object.
String	<code>getCursorName()</code> Retrieves the name of the SQL cursor used by this <code>ResultSet</code> object.
Date	<code>getDate(int columnIndex)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a <code>java.sql.Date</code> object in the Java programming language.
Date	<code>getDate(int columnIndex, Calendar cal)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a <code>java.sql.Date</code> object in the Java programming language.
Date	<code>getDate(String columnLabel)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a <code>java.sql.Date</code> object in the Java programming language.
Date	<code>getDate(String columnLabel, Calendar cal)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a <code>java.sql.Date</code> object in the Java programming language.
double	<code>getDouble(int columnIndex)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a double in the Java programming language.
double	<code>getDouble(String columnLabel)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a double in the Java programming language.
int	<code>getFetchDirection()</code> Retrieves the fetch direction for this <code>ResultSet</code> object.
int	<code>getFetchSize()</code> Retrieves the fetch size for this <code>ResultSet</code> object.
float	<code>getFloat(int columnIndex)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a float in the Java programming language.
float	<code>getFloat(String columnLabel)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as a float in the Java programming language.
int	<code>getHoldability()</code> Retrieves the holdability of this <code>ResultSet</code> object.
int	<code>getInt(int columnIndex)</code> Retrieves the value of the designated column in the current row of this <code>ResultSet</code> object as an int in the Java programming language.

Imagen 30. Documentación para ResultSet.

Fuente: Desafío Latam.

En la imagen se muestra una pequeña muestra de todos los métodos que ofrece la interfaz `ResultSet`. <https://docs.oracle.com/javase/8/docs/api/java/sql/ResultSet.html>

PreparedStatement

Interface que representa una sentencia sql precompilada. La sentencia sql es almacenada en el `preparedStatement` y gracias a ella es posible ejecutarla para trabajar con sentencias select, update, delete, etc.

```
PreparedStatement pstmt = con.prepareStatement("UPDATE EMPLOYEES
                                                SET SALARY = ? WHERE ID = ?");
pstmt.setBigDecimal(1, 153833.00)
pstmt.setInt(2, 110592)
```

Imagen 31. PreparedStatement.

Fuente: Desafío Latam.

Connection

Una conexión (sesión) con una base de datos específica. Las instrucciones SQL se ejecutan y los resultados se devuelven dentro del contexto de una conexión. La base de datos de un objeto puede proporcionar información que describe sus tablas, su gramática SQL admitida, sus procedimientos almacenados, las capacidades de esta conexión, etc. Esta información se obtiene con el método *getMetaData*.



Nota: Al configurar a *Connection*, las aplicaciones JDBC deben usar el método de conexión apropiado como *setAutoCommit* o *setTransactionIsolation*. Las aplicaciones no deben invocar comandos SQL directamente para cambiar la configuración de la conexión cuando hay un método JDBC disponible. Por defecto, un objeto *Connection* está en modo de confirmación automática, lo que significa que confirma automáticamente los cambios después de ejecutar cada instrucción. Si se ha deshabilitado el modo de confirmación automática, se debe llamar explícitamente al método para confirmar los cambios; de lo contrario, los cambios en la base de datos no se guardarán.

Drivers de conexion con Oracle

En las siguientes líneas se describen los tipos de drivers de conexion con oracle:

Oracle JDBC Thin Driver

- Es la implementación recomendada por Oracle para la mayoría de requerimientos de conexión.
- El driver JDBC Thin está compuesto en su totalidad de código java y es conocido como un driver tipo IV.
- Soporta desde la versión 5 de java y es independiente de la plataforma y no requiere de la instalación de software adicional de Oracle para desarrollos del lado del cliente.
- El tipo *thin* proporciona conexión directa a la base de datos gracias a la implementación pura en lenguaje java.
- El driver soporta el protocolo TCP/IP y puede ser ejecutado en cualquier máquina que tenga instalada la java virtual machine.

Si se desea acceder a las funcionalidades propias de esta implementación debe dirigirse al package oracle.jdbc.

Oracle JDBC OCI Driver

- El driver JDBC OCI es conocido como driver de tipo II.
- Su característica es que necesita de la instalación de un cliente oracle en la máquina en la cual se implemente.
- Se utiliza para implementaciones empresariales con ciertas características especiales las cuales escapan al alcance de este curso.

Para entender y aplicar lo anteriormente expuesto, a continuación se generará un pequeño proyecto java estándar que se encargará de conectar con una tabla de una base de datos creada en Oracle 18c express. El programa se puede dividir en 3 partes o pasos:

- Levantar el driver y establecer la conexión.
- Ejecutar la query, recorrerlo y mostrar los datos por pantalla
- Cerrar la conexión.

Descarga del driver Oracle JDBC

Para poder conectar cualquier programa Java con una base de datos, se necesita el driver de conexión.

1. Para el caso de la tecnología Oracle debe dirigirse a la página oficial de descarga:
<https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html>
2. Descargar el driver ojdbc8.jar.
3. Guardar en una ubicación conocida porque se ocupará más adelante.