



Módulo Proyecto

Proyecto Grupal

Módulo
Persistencia de Datos y Transaccionalidad
Nombre de la entrega
Implementación de tecnologías sockets, persistencia de datos y transacciones
Nivel académico
Profesional
Tipo de entrega
Proyecto grupal

INSTRUCCIONES PARA REALIZAR LA ENTREGA

Definición del proyecto

El proyecto se puede desarrollar, utilizando sockets con Java, C++ o Python sobre Windows o Linux utilizando la base de datos MySQL, PostgreSQL, Hsqldb, MariaDB u Oracle 11gR2 express, que son motores de bases de datos libres.

El objetivo es aprender, en primera instancia, a trabajar con aplicaciones transaccionales, para lo cual se debe disponer de un portátil con Linux o Windows que tenga instalado el lenguaje de computación que use el estudiante a diario a través de Eclipse o NetBeans, pero que tenga las librerías necesarias para el manejo de sockets y un motor de base de datos.

En segunda instancia, se busca que se aprenda a instalar un motor de base de datos en Linux o Windows, luego desarrollar un modelo de base de datos que se explica más adelante para la segunda entrega y cargar datos.

Se sugiere que cada estudiante elabore los talleres en su portátil para que adquiera el conocimiento y luego integre los resultados en el documento maestro.

Se pide elaborar un **DOCUMENTO COMPLETO** (con todo el proceso de las especificaciones correspondientes que se describen a continuación en cada requerimiento de la respectiva entrega).

Debe usar normas APA, para la elaboración del documento. .

El estudiante debe ir elaborando las entregas 1, 2 y 3 en cada semana conforme a las instrucciones para la elaboración del proyecto.

Para elaborar el proyecto, el estudiante debe ir aplicando los procesos metodológicos del programa, hacer ejercicios de clase y una vez afianzado el conocimiento, debe construir las respectivas entregas en las fechas pactadas en el curso.

NOTA: elaborar el código en java, Python o en C, en las entregas respectivas de las semanas 3 y 5. **Cada entrega es independiente una de la otra en documento aparte.**

Desarrollo de un chat bidireccional usando sockets

Realice el siguiente paso a pasos en su computador:

1. Cree una cuenta en Oracle. Ver la referencia, aquí, [1]:
<https://login.oracle.com/mysso/signon.jsp>
2. Baje el instalador Java de la página oficial de Oracle., versión 11.0.12 o superior de [2], aquí:
[jdk-11.0.12-windows-x64_bin.exe](#)
3. Siga las instrucciones del manual de instalación que están en [2].
4. Baje el Eclipse de aquí [3].
5. Instale la versión de eclipse que se muestra en la imagen dada a continuación.

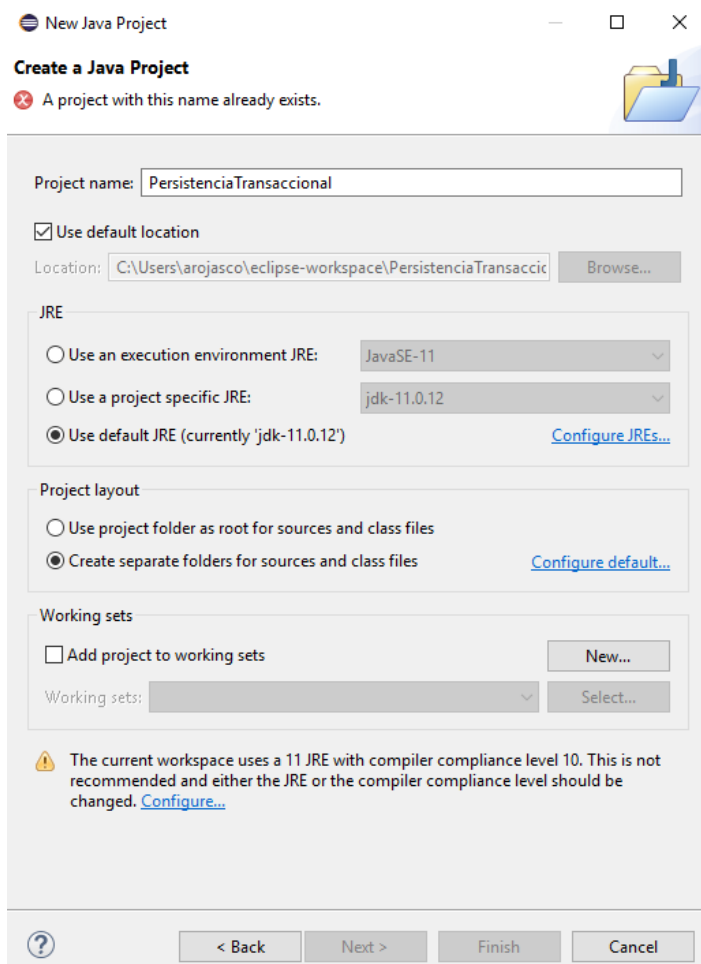


**Instala esta
versión**

Figura 1. Instalación de Eclipse

Fuente: ejecución del instalador de Eclipse

6. Cree el proyecto de persistencia como se muestra en la figura 2.



Elija el compilador de Java 11+

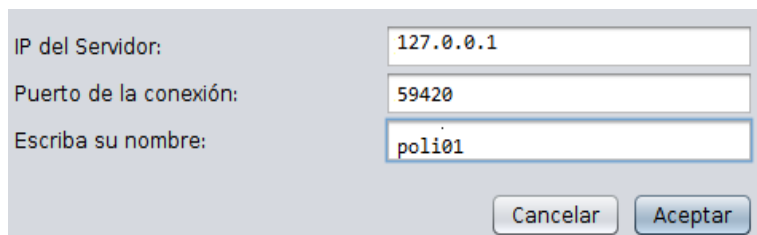
Figura 2. Creación del proyecto

Fuente: elaboración propia

Realice los siguientes pasos en la preparación de la entrega:

7. Haga un marco teórico con los fundamentos de programación con sockets.
8. Diseño del modelo de comunicaciones con sockets, defina las IP's que va a utilizar y el puerto a usar para el programa Servidor.
9. Desarrolle la primera parte del proyecto que consiste en la construcción de un sistema de comunicaciones vía chat bidireccional con sockets entre dos programas, uno que hace las veces de servidor y otro que hace las veces de cliente.
10. El Servidor se debe iniciar y estar atento a los llamados de cliente.

11. El cliente arranca e inicia la conversación con el servidor, dando sus credenciales, con los datos para establecer conectividad (en este ejemplo la ip del *localhost*, el puerto y su nombre de usuario) , como se muestra en la figura 3, pero se puede hacer por consola, sin necesidad de interfases gráficas (queda a su gusto montar interfases gráficas)



A screenshot of a graphical user interface window for a client application. The window has a light gray background and contains three input fields with labels to their left. The first field is labeled 'IP del Servidor:' and contains the text '127.0.0.1'. The second field is labeled 'Puerto de la conexión:' and contains the text '59420'. The third field is labeled 'Escriba su nombre:' and contains the text 'poli01'. Below the input fields are two buttons: 'Cancelar' and 'Aceptar'.

Figura 3. Ventana del cliente para establecer conectividad con el servidor

Fuente: elaboración propia

12. El servidor va mostrando la lista de clientes conectados, (poli01, poli02, etc., en la medida que se vayan conectando usuarios por el programa Cliente. Una vez usted se identifique y el servidor registre el usuario, le devuelve el log de usuarios conectados y usted elige uno con el que quiere conversar. Ver figura 4.

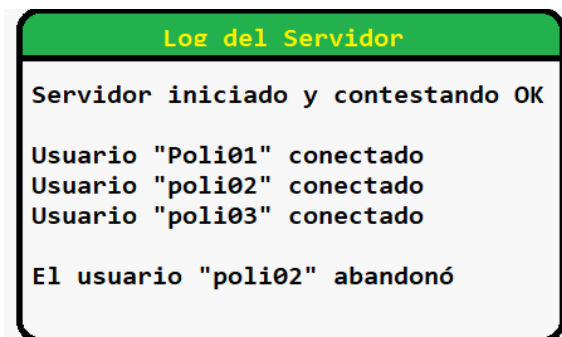


Figura4. Log del servidor

Fuente: elaboración propia

13. Usted elige un usuario para chatear y cuando quiera abandonar envía un mensaje de “chao” para que el servidor, marque a dicho usuario como que abandonó el chat y ahí termina el proceso de comunicaciones de esta entrega. En la figura 5, se simula como deben aparecer las conversaciones.

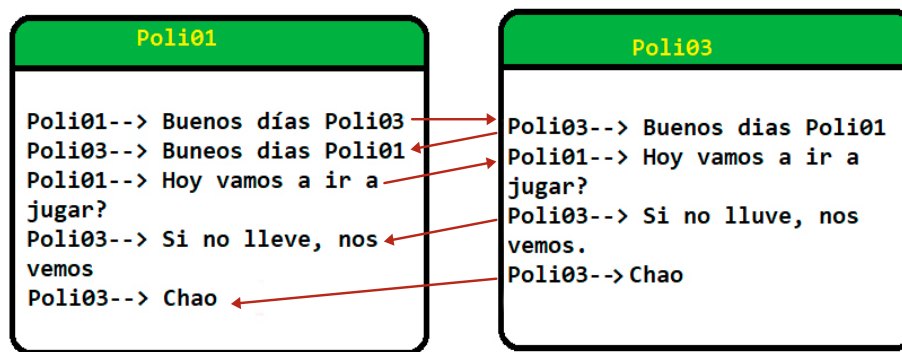


Figura 5. Charla entre usuarios a través del servidor

Fuente: elaboración propia

Realice los siguientes pasos en la preparación de la entrega:

1. Explicación del código fuente de los sockets y muéstrelo funcionando.
2. Suba el video a YouTube y recuerde verificar que funciona.
3. Adjunte el enlace del video al documento de la entrega en una parte visible al final, antes de las conclusiones.

Realice los siguientes pasos en la preparación de la entrega:

1. Incluya en el documento la información resumida del estado del arte de lo que es un socket.
2. Incluya la explicación de las líneas de código de los sockets construidos para el taller.
3. Agregue el diseño de la red utilizada para el desarrollo de su aplicación y la trazabilidad con las pruebas del proceso descrito anteriormente.
4. Adjunte el enlace del video realizado
5. **Adjunte el código fuente a la entrega** con conclusiones y bibliografía consultada.
6. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.

ENTREGA 2 SEMANA 5

Diseño de un modelo ER de base de datos de Recursos Humanos y cargar al menos cinco datos de prueba en cada tabla. Empiece cargando los datos de las tablas padres, luego llene las tablas hijas y por último cargue los datos de las tablas nietas, bisnietas, etc.

Realice el siguiente paso a paso en su computador:

- » Montar un motor de base de datos y cargar 5 registros en cada tabla de manera manual. Para ello, realice los siguientes pasos:
 - a. Con la cuenta de usuario creada en www.oracle.com para la entrega 1, puede bajar la versión de "Oracle Database 18c Express Edition for Windows x64". Igual puede usar cualquiera de los otros motores de bases de datos libres.
 - b. Para el caso Oracle, la puede bajar de [4]. Luego de haberse registrado, puede comenzar la instalación, como sigue:
 - c. Instalación del gestor de Base de Datos Oracle como se muestra en la figura 6.

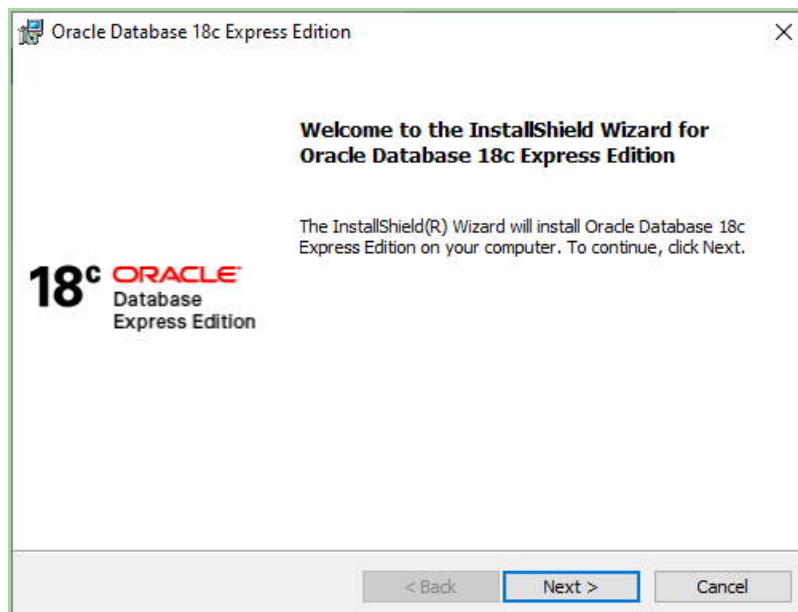


Figura 6. Inicie la instalación del Gestor de BD

Fuente: ejecución del proceso

- d. Se debe ir avanzado con next en los pasos anteriores hasta llegar al cuadro de diálogo donde se pide la clave. La clave debe ser escrita en un Bloc de notas, para verificar que quede correcta y no tenga caracteres extraños. Secopia y se pega como se muestra en la figura 7.

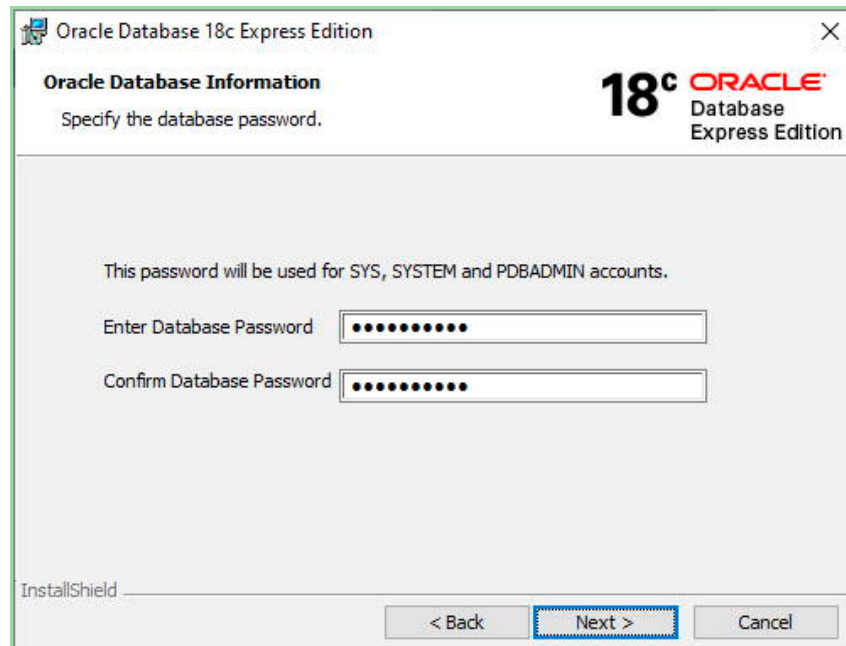


Figura 7. Ingreso de clave del System

Fuente: ejecución del proceso

- e. En instantes, el gestor queda instalado.
- f. Baje el entorno *visual free* de "Oracle SQL Developer" de [5]., para comunicarse con la base de datos
- g. Descargue el archivo y proceda a descomprimir el archivo zip en una carpeta (este programa no requiere instalación). Ver figura 8

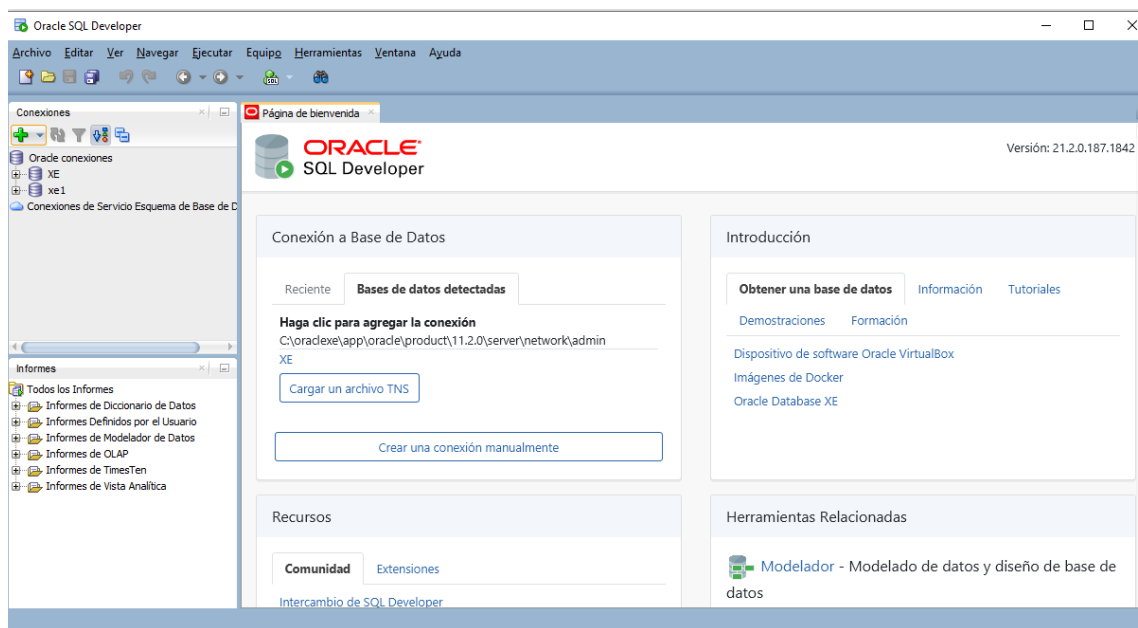


Figura 8. Oracle SQL Server para ejecutar

Fuente: ejecución del proceso

- h. Cree un acceso directo y colóquelo en el escritorio.
- i. Con el usuario *System*, Cree un *tablespace* llamado DATOS. Cree un usuario llamado Poli01 que use por defecto el *tablespace* Datos y como temporal, el *tablespace temp*. Asigne los permisos de *connect*, *resource* y *session*.

Asigne los permisos así:

```
GRANT CREATE SESSION TO Poli001;
GRANT CONNECT TO Poli001;
GRANT RESOURCE TO Poli001;
```

Permisos para usuario de modificación de datos (permiso sobre objeto):

```
GRANT SELECT, INSERT, UPDATE, DELETE ON T_PEDIDOS TO Poli001;
```

Desconéctese como *System* y entre como Poli01. Comience a trabajar.

Realice los siguientes pasos en la preparación de la Segunda Entrega:

- » Use el siguiente modelo conceptual de la figura 9, para crear el modelo lógico y el modelo físico
- » Genere las tablas del modelo físico en el gestor de base de datos

» Cargue 5 datos de prueba por tabla

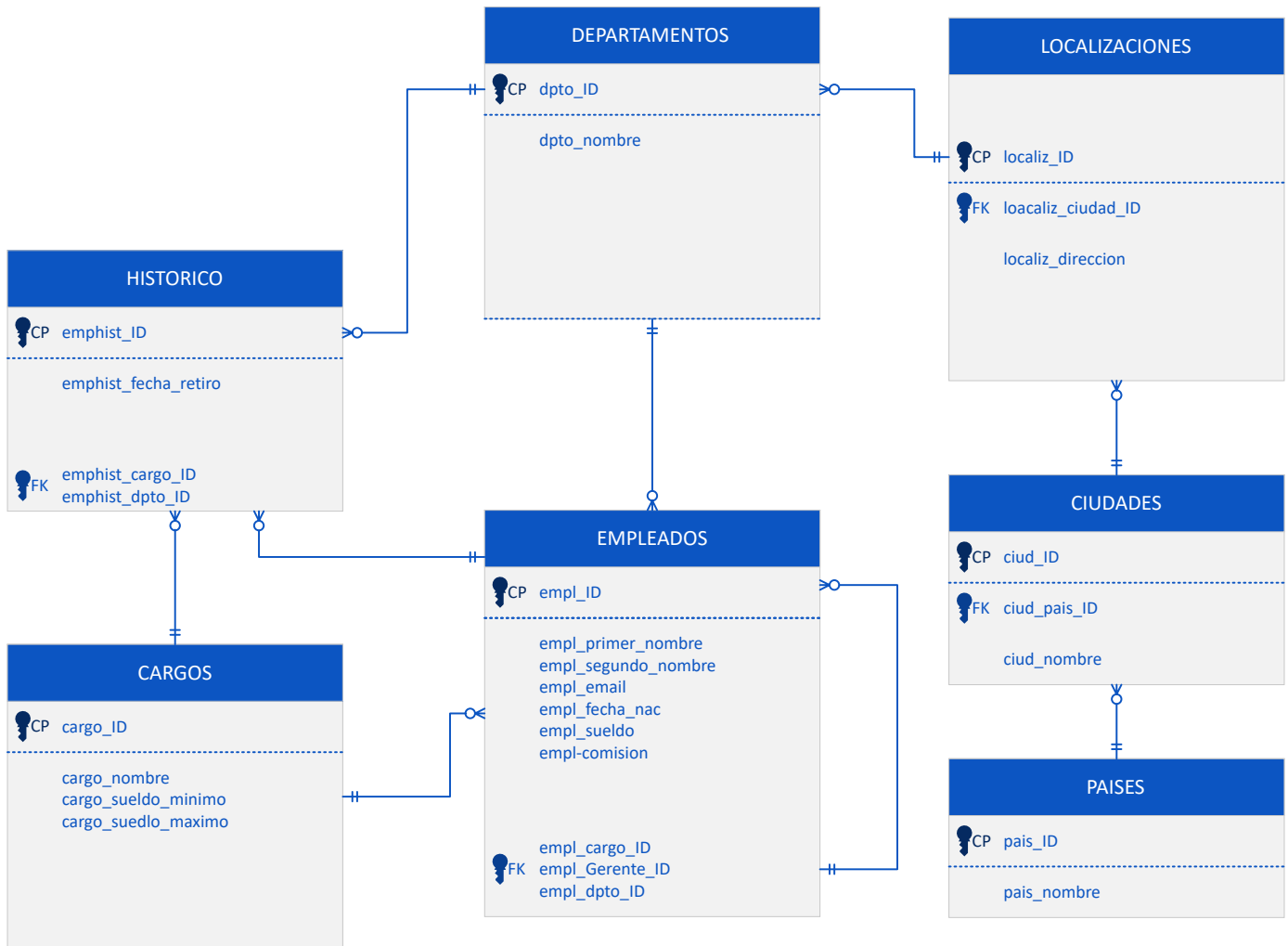


Figura 9. Modelo de Base de datos Relacional de Recursos Humanos

Fuente: elaboración propia

Construya un vídeo de sustentación en el que incluya

1. Un video donde explique el modelo lógico (es decir, como resuelve las relaciones muchos a muchos) y explique el código del del modelo físico, en el momento de crear las tablas, con sus llaves, atributos y referencias.
2. Suba el video a YouTube y pruebe su funcionamiento antes de colocar el enlace en el documento de la segunda entrega. Adjunte el enlace del video al informe.

Desarrolle el informe teniendo en cuenta los siguientes elementos

1. Elabore el documento con la información resumida del estado del arte de lo que, de las bases de datos, el modelo conceptual, el modelo lógico y el modelo físico.
2. Incluya pantallazos donde se muestre el proceso de creación de las tablas, los índices y las referencia.
3. Adjunte el código fuente de los scripts de creación de las tablas y la carga de 5 datos por tabla.
4. Adjunte el enlace del video al final del documento antes de las conclusiones.
5. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.

ENTREGA 3 Y SUSTENTACIÓN SEMANAS 7 Y 8

Acceder a la base de datos usando sockets para hacer las operaciones transaccionales

1. Elabore el socket Sever que reciba peticiones del socket Cliente y resuelva dichas peticiones contra la base de datos de Recursos Humanos, usando cualquiera de los lenguajes de programación como Java, Python, C o C++.
2. Elabore un programa Socket Cliente que use la ip del Socket Servidor y el puerto del Servidor para conectarse al Socket servidor y enviar transacciones.
3. El *Socket Cliente* captura la información del usuario y el *socket Server* debe realizar las siguientes operaciones.
 - a. *Insert* --- > para Insertar un empleado en la base de datos
 - b. *Update* --- > Para actualizar los datos de un empleado en la base de dagos
 - c. *Select* --- > Para consultar un empleado en la base de datos
 - d. *Delete* --- > Borrar un empleado de la base de datos, previa inserción de la tabla Históricos.

4. Los resultados deben ser ejecutados en la base de datos por el socket Server, pero visualizados solamente en el socket cliente.
5. ¿Cómo hacerlo?
 - a. Cree un país, una ciudad, una localización un departamento, un cargo y un empleado. Agregue el proceso al documento.
 - b. Haga una consulta de un empleado y agregue la consulta al documento
 - c. Ejecute un retiro de un empleado marcándolo como borrado en un campo de la tabla empleados y agréguelo al histórico.
 - d. Ejecute un cambio de dirección y ciudad de un empleado y luego la consulte para ver si quedó correcto el cambio y agregue los resultados de la operación al documento.
6. Agregar al documento el código de cada operación y explicar que hace cada programa (documentar el programa).
7. El proyecto debe realizarse en grupos de 3, 4 o 5 estudiantes.

Para la entrega final, tenga en cuenta los siguientes pasos:

Elabore el documento con la información resumida del estado del arte de lo que es un proceso transaccional, junto con su explicación de lo que usted cree que hace el sistema de transacciones.

- h. Adjunte el código fuente a la entrega de los programas Socket Server y Socket cliente, junto con la explicación de lo que hace en cada programa.
- i. Construya un video donde explique el código y muestre el funcionamiento.
- j. Suba el video a YouTube y pruebe su funcionamiento antes de pegarlo al documento
- k. Adjunte el enlace del video a la entrega.
- l. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.

NOTA: el documento final debe contener todo lo requerido en la tercera entrega.

Referencias para apoyar el desarrollo de la primera entrega:

[1] Sitio para crear cuenta en Oracle. <https://login.oracle.com/myssso/signon.jsp>

[2] Sitio Oracle para bajar el software Java, versión 11.0.12 - JDK (Versión Libre). <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

[3] Sitio para bajar el eclipse. <http://www.eclipse.org/downloads/>

Referencias para apoyar el desarrollo de la segunda entrega:

[4] Gestor de Base de datos Oracle. <https://www.oracle.com/database/technologies/xe-downloads.html>

[5] Entorno visual para Oracle. <https://www.oracle.com/database/technologies/appdev/sqldeveloper-landing.html>

[6] Asignación de privilegios de usuario. <https://ora.u440.com/usuarios/grant.html>