



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

Miércoles 14 de agosto de 2024

Hora de inicio:

17:20

Sistemas de Bases de Datos 2

Diego André Mazariegos Barrientos

CLASE 4

PROCEDIMIENTOS, FUNCIONES Y TRIGGERS CON TRANSACCIONES

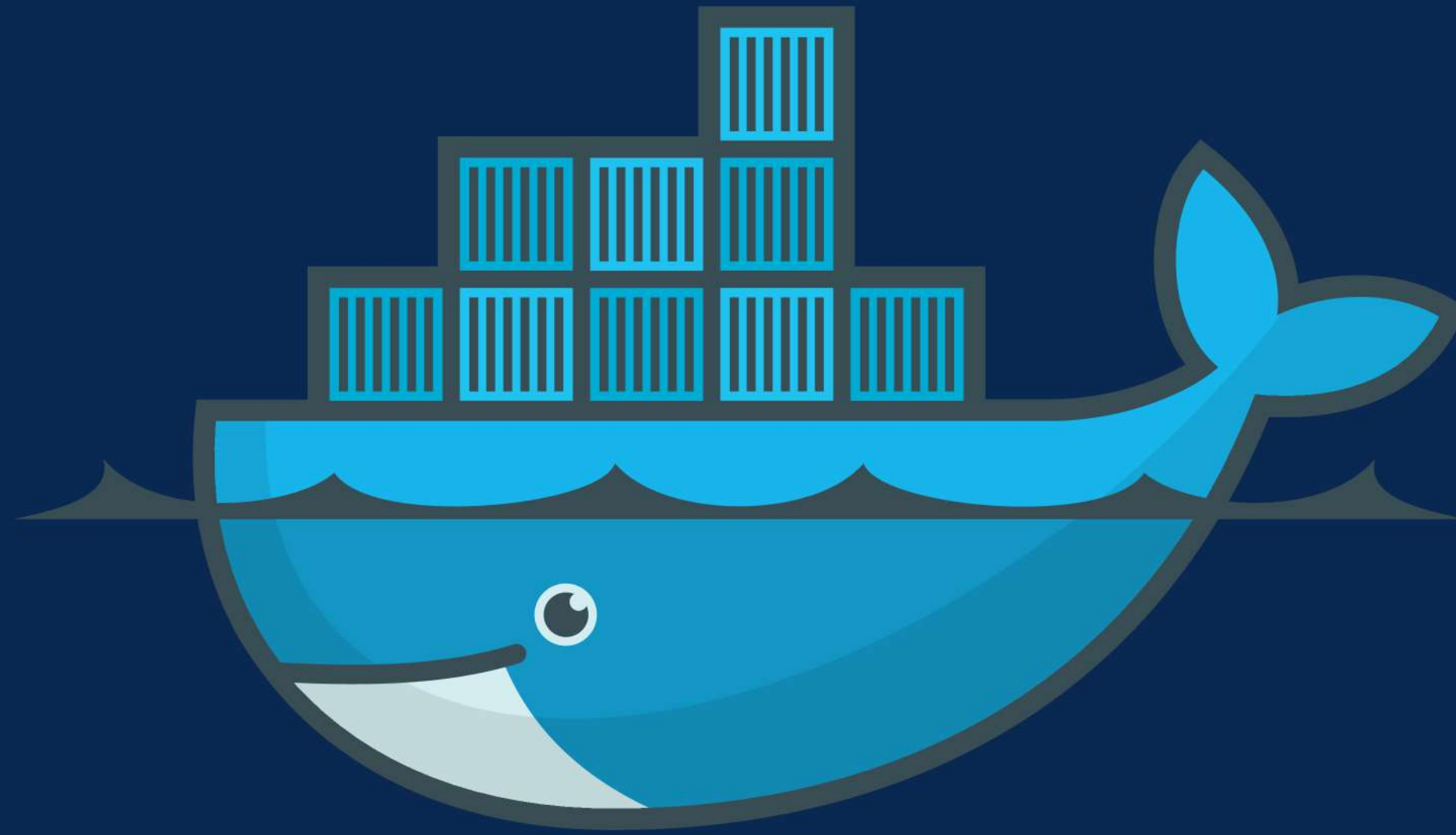
SISTEMAS DE BASES DE DATOS 2

AGENDA

- Datos extras
 - Docker explicación
- PL/SQL
 - Introducción
 - SP's y PL/SQL
 - PL/pgSQL
- TSQL
 - Transacciones y TSQL
- Triggers
- Ejemplos prácticos

Repaso clase
anterior y extras

DOCKER



<https://www.udemy.com/course/guia-completa-de-docker-kubernetes-con-spring-boot>

Qué es Docker?

Docker es una plataforma de contenedores, una herramienta para crear y administrar contenedores.

Container



Es una caja de herramienta asilada con un porción de un sistema operativo para ejecutar aplicaciones en nuestra maquina host.

Es un empaquetado del código y dependencias para ejecutar que ese código, la aplicación (ejemplo aplicación con Spring + JDK runtime).

Un mismo contenedor que se ejecuta siempre debe reproducir exactamente el mismo comportamiento de la aplicación, sin importar dónde o quién lo ejecuta

Por qué contenedores?

¿Por qué necesitaríamos empaquetar aplicaciones aisladas e independientes en contenedor en el desarrollo de aplicaciones?



Microservicios,
escalamiento y alta
disponibilidad

Nos permite publicar
nuestra aplicación en
mas de una instancia
ya sea en diferentes
puertos o en
diferentes ip o
maquinas

Diferentes ambientes de desarrollo
y producción

Podríamos necesitar ejecutar
nuestros test y probar nuestra
aplicación en el mismo ambiente de
producción en que se ejecutará mas
adelante

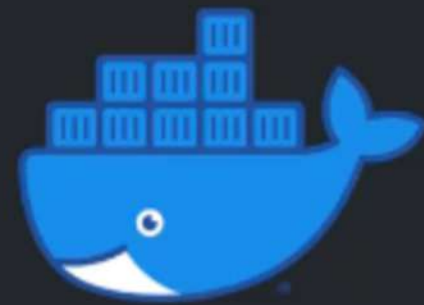
Diferentes ambientes de desarrollo
y versiones en un equipo de
desarrollo

Cada desarrollador del equipo
debería de tener exactamente el
mismo ambiente y versiones
cuando trabajan en el mismo
proyecto

Conflictos de versiones y
herramientas entre los diferentes
proyectos

Cuando trabajamos en diferentes
proyectos y hacemos un switch
entre uno y otro podríamos tener
conflicto entre las versiones

Arquitectura de Contenedores



Contenedores



Contenedores vs Maquinas Virtuales

Contenedores

Encapsula una fina capa de S.O, el ambiente y la aplicación a ejecutar

Optimizado, consumo mínimo de recursos, poco espacio en disco, muy rapidos

Muy simple de compartir, distribuir, construir y de ejecutar

Maquinas virtuales

Contiene un completo sistema operativo, duplicación redundante en vez de lo justo y necesario

Gran consumo de recursos del sistema, bajo rendimiento, lentos, ocupa mucho espacio en disco

Compartir, distribuir, construir puede ser mas complicado por su tamaño y configuraciones

Imágenes vs Contenedores

Imágenes



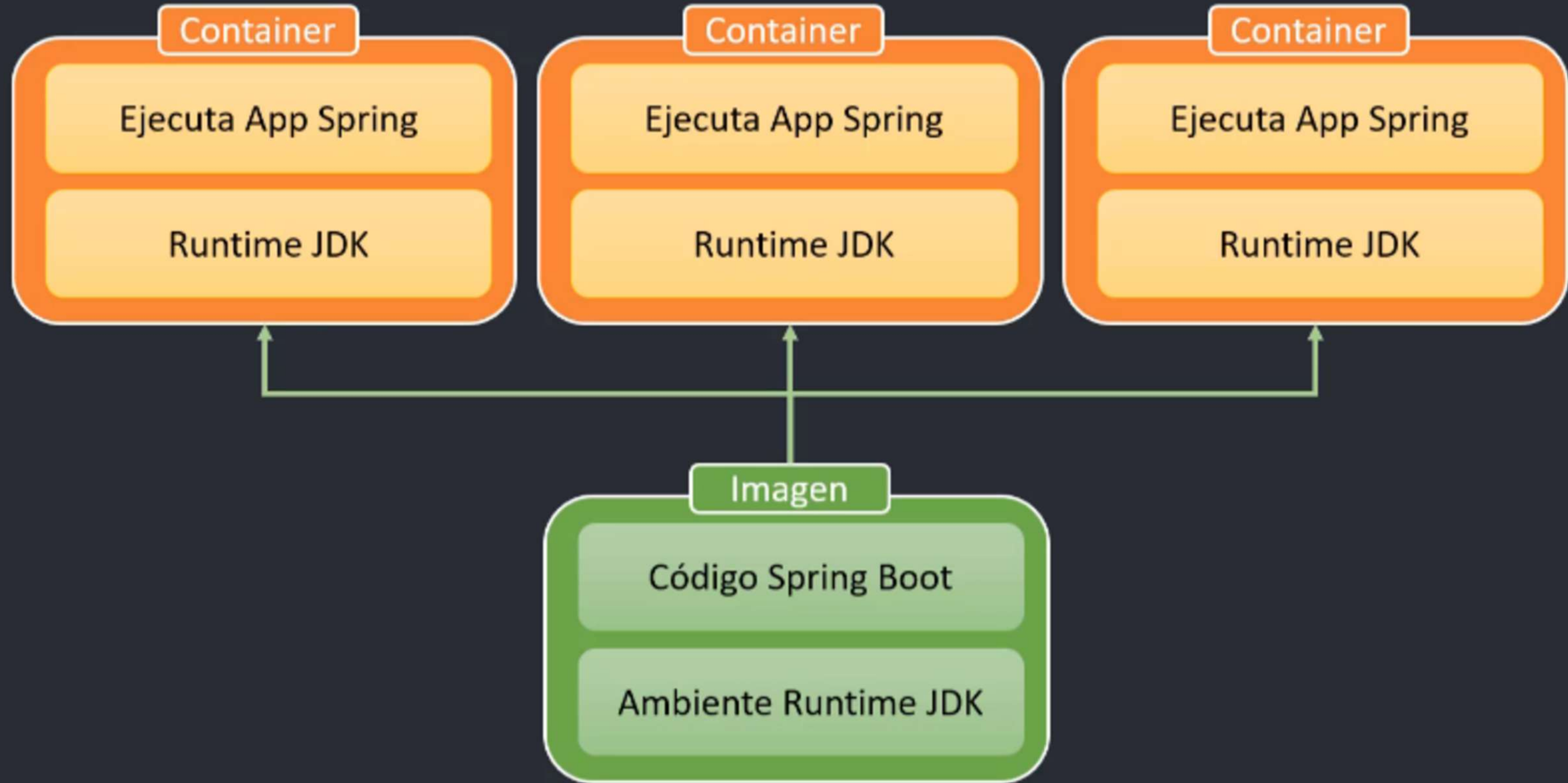
Contenedores

Plantillas para ejecutar contenedores

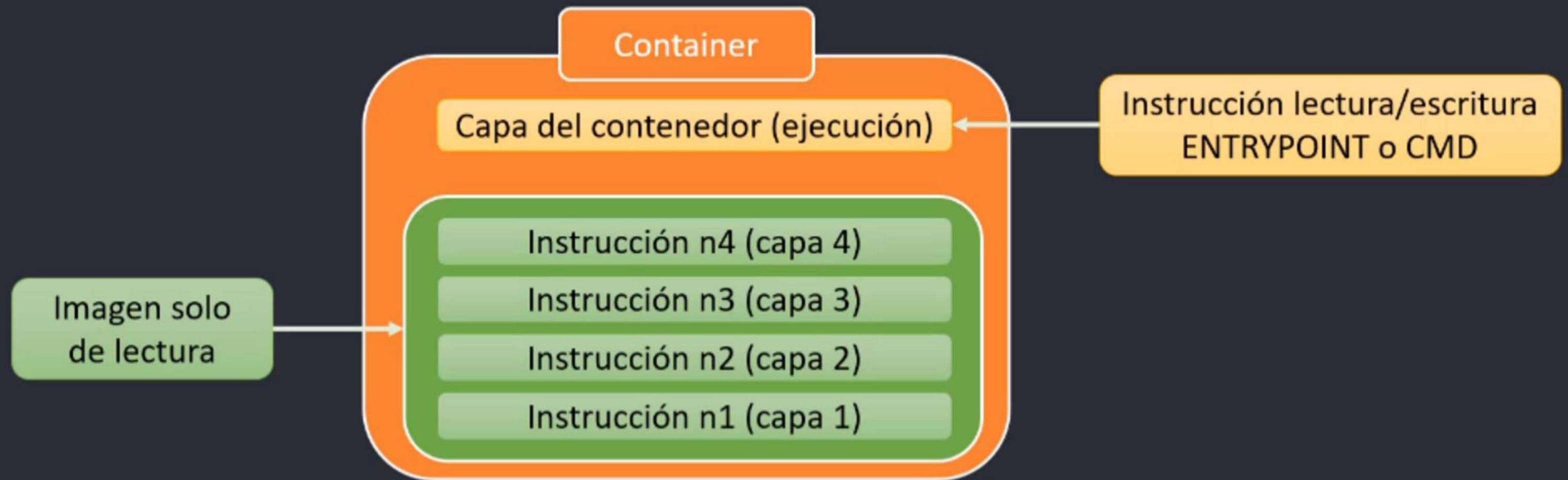
Contiene el código y ambiente necesario
(runtime, herramientas, librerías)

Instancia que ejecuta la aplicación y el
ambiente necesario para su ejecución

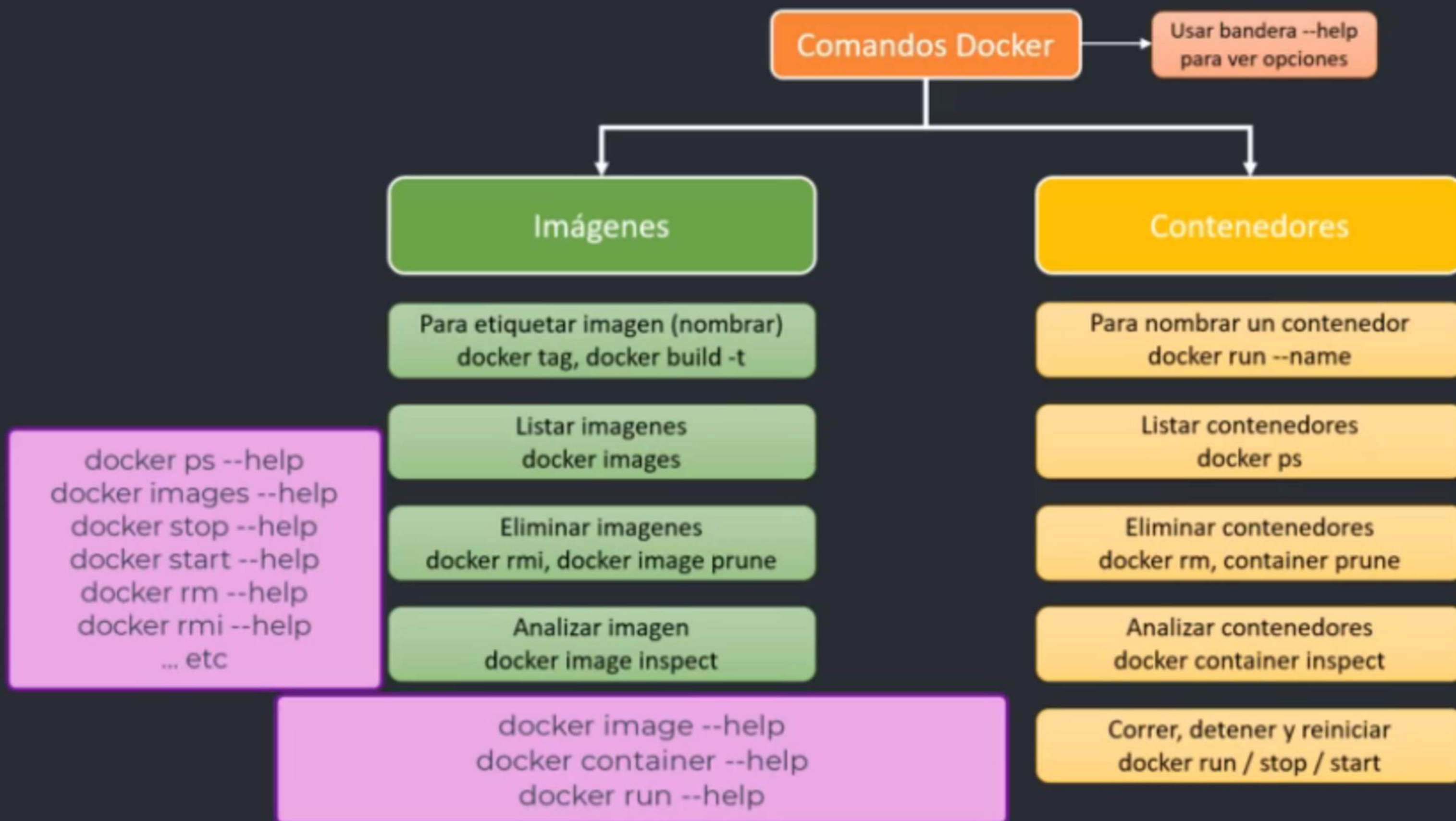
Imágenes vs Contenedores



Capas de la imagen



Comandos: imágenes y Contenedores





Omkar Srivastava

Helps job aspirants to crack FAANG product companies

Docker Concept

DOCKER CONTAINER

A running instance of a Docker image.

DOCKERFILE

Instructions to build a Docker image.

DOCKER IMAGE

A standalone package with everything needed to run an application.

DOCKER SECRETS

Securely manage sensitive data, such as passwords and API keys.

DOCKER SWARM

Native clustering and orchestration tool for Docker.

DOCKER REGISTRY

A repository for storing and distributing Docker images.

DOCKER CLI

The command-line interface to interact with Docker.

DOCKER NETWORKS

Enables communication between containers and the host system.

FORO SEMANA #4

PL/SQL
T-SQL

Definición

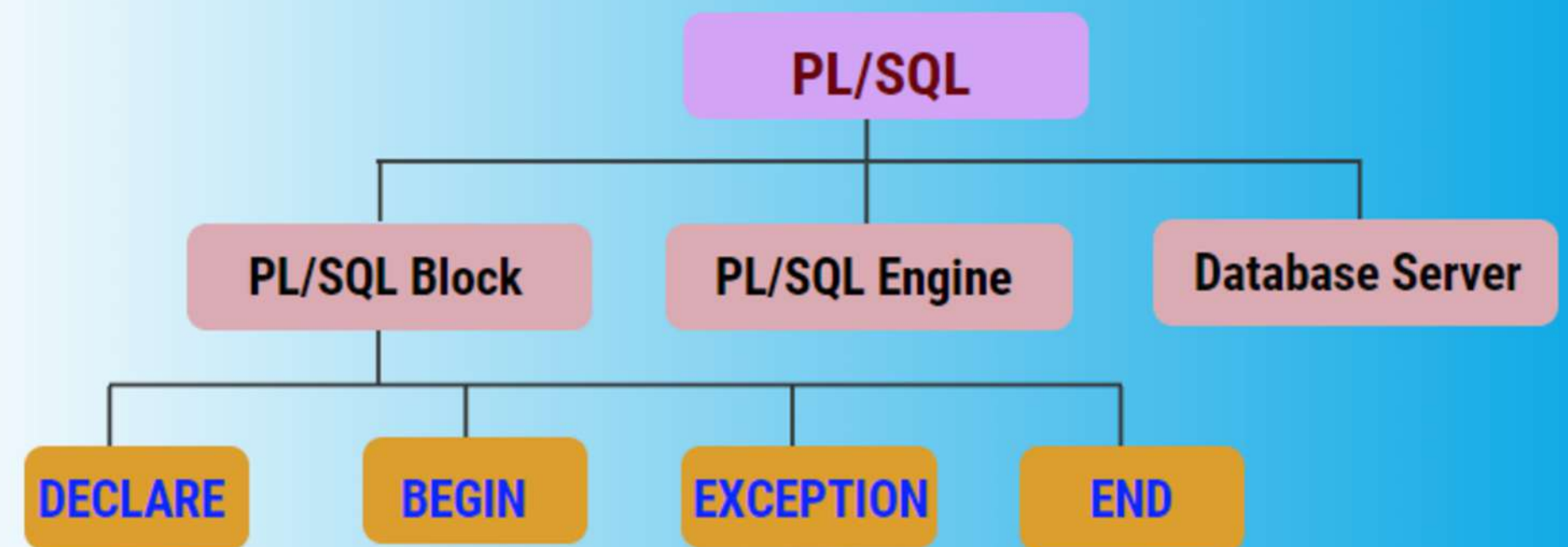
PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación procedural extendido para SQL en bases de datos Oracle.

¿Qué es
PL/SQL?

Características

- Integración con SQL: permite ejecutar SQL dentro de bloques de PL/SQL.
- Portabilidad: los programas PL/SQL son portables entre diferentes plataformas Oracle.
- Manejo de Excepciones: ofrece un manejo robusto de errores y excepciones.

What is PL/SQL?



Definición

T-SQL (Transact-SQL) es una extensión de SQL desarrollada por Microsoft y Sybase, utilizada principalmente en Microsoft SQL Server y en bases de datos de Sybase ASE. T-SQL amplía el estándar SQL con características adicionales que permiten realizar operaciones más complejas y procedurales directamente en la base de datos.

Características

- Procedimientos almacenados
- Funciones definidas por el usuario
- Manejo de Errores
- etc ...

¿T-SQL (Transact-SQL)?



y PostgreSQL?

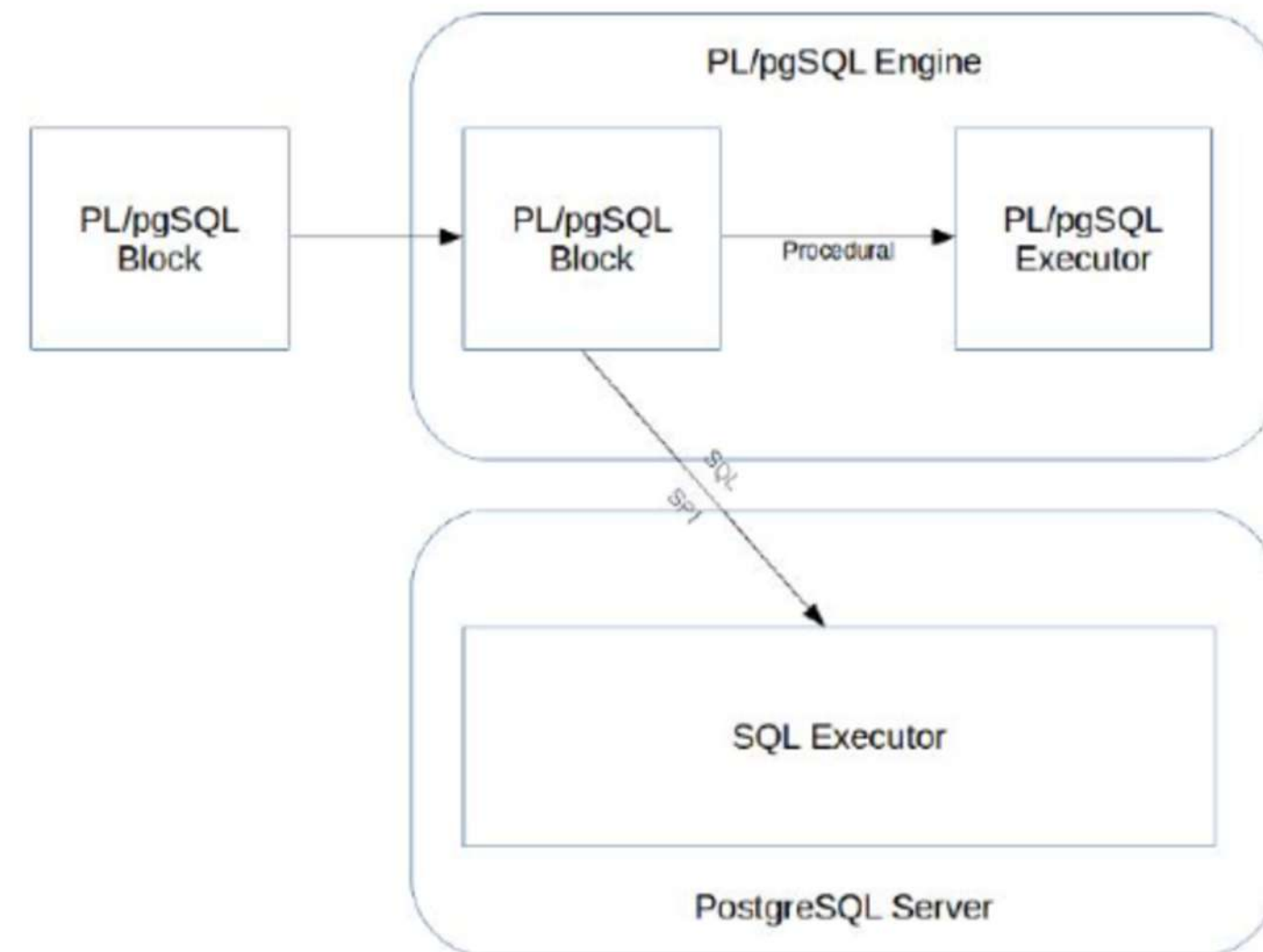
Definición

PL/pgSQL en PostgreSQL ofrece funcionalidades comparables a PL/SQL, como la capacidad de escribir procedimientos almacenados, funciones, control de flujo, manejo de excepciones, y más. Aunque hay similitudes, PL/pgSQL y PL/SQL no son intercambiables directamente, y existen diferencias en la sintaxis y el comportamiento de ciertos comandos y funciones.

Características

- Uso de SQL embebido
- Estructuras de Control
- Manejo de Excepciones
- Procedimientos y Funciones
- Triggers
- Declaración de Variables
- Transacciones

¿Qué es PL/pgSQL?



Funciones
Procedimientos
Triggers

Definición

Una función en PostgreSQL es un bloque de código que puede tomar parámetros de entrada, realizar operaciones y devolver un valor (o conjunto de valores). Las funciones están diseñadas para ser utilizadas en consultas SQL, lo que permite ejecutar lógica compleja directamente en las operaciones de selección, inserción, actualización, etc. Las funciones en PostgreSQL pueden devolver tipos de datos simples, filas, tablas o valores específicos.

¿Qué es una función?



PLPGSQL (FUNCIONES)

- PUEDE EXISTIR MUCHAS CON EL MISMO NOMBRE PERO CON DISTINTA FIRMA.
- LLAMADAS DESDE CLP CON “SELECT <FUNCTION>”
- CREACIÓN /ELIMINACIÓN

```
CREATE [OR REPLACE] FUNCTION <name> ( [argtype , [...]] )  
RETURNS <return_type>  
AS ' definition '  
LANGUAGE plpgsql  
  
DROP FUNCTION <name> ( [argtype , [...]] )
```




PLPGSQL (FUNCIONES)



- EJEMPLO:

```
CREATE FUNCTION SUMA (INT,INT,INT,INT)
```

```
RETURNS INT AS \
```

```
DECLARE
```

```
    SUMA INT;
```

```
BEGIN
```

```
    SUMA:=$1+$2+$3;
```

```
    SUMA:=SUMA-$4;
```

```
    RETURN SUMA;
```

```
END;
```

```
\ LANGUAGE plpgsql;
```

```
sampledb=> SELECT SUMA (1,2,3,4);
```

```
Suma
```

```
-----
```

```
2
```

Declaración
de variable

Cuerpo



PLPGSQL (FUNCIONES)

- EJEMPLO COMPLEJO:

```
CREATE FUNCTION VERIFICAR (INT,INT,INT) RETURNS BOOLEAN AS'  
DECLARE  
    MAX INT;  
    MIN INT;  
    RESUME INT;  
BEGIN  
    SELECT INTO RESUME SUM(SCORE) FROM SALES;  
    IF( RESUME IS NULL) THEN  
        RETURN FALSE;  
  
    END IF;  
    MAX:=1;  
    WHILE (RESUME < MAX) LOOP  
        MIN:=SUMA(MAX,RESUME,RESUME+MAX,10);  
        RESUME:=RESUME-1;  
        IF( MIN % 2 <> 0) THEN  
            RETURN TRUE  
        END IF;  
    END LOOP;  
    RETURN FALSE;  
END;  
'LANGUAGE plpgsql;
```


Definición

Un procedimiento en PostgreSQL es un bloque de código que puede ejecutar operaciones en la base de datos, como modificar tablas, insertar registros, actualizar datos, etc. A diferencia de las funciones, los procedimientos no están diseñados para ser utilizados dentro de consultas SQL y no devuelven valores directamente. En cambio, se utilizan principalmente para realizar tareas que afectan el estado de la base de datos y pueden incluir lógica transaccional, como COMMIT y ROLLBACK.



¿Qué es un
procedimiento?

```
CREATE PROCEDURE insertar_cliente(nombre VARCHAR, edad INTEGER)
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO clientes(nombre, edad) VALUES (nombre, edad);
END;
$$;
```

```
CALL insertar_cliente('Juan Perez', 30);
```




PLPGSQL (TRIGGERS)

- VARIABLE PRE-DEFINIDAS

Nombre	Tipo	Descripción
NEW	%ROWTYPE	Nuevos Valores (INSERT UPDATE)
OLD	%ROWTYPE	Valores Antiguos (UPDATE DELETE)
TG_NAME	NAME	Nombre del TRIGGER
TG_WHEN	TEXT	BEFORE AFTER
TG_LEVEL	TEXT	ROW SENTENCIA (SQL)
TG_OP	TEXT	INSERT, UPDATE DELETE
TG_RELID	OID	Identificador de la tabla
TG_RELNAME	NAME	Nombre de la tabla
TG_NARGS	INT	Numero de argumentos
TG_VARGS	TEXT[]	Argumentos



PLPGSQL (TRIGGERS)

- EJEMPLO:

```
CREATE FUNCTION UPDATE_FORMS_NUMDOWN()  
RETURNS TRIGGER AS '  
BEGIN  
    UPDATE TEST SET  
        FORMS=FORMS-1 WHERE TESTID=OLD.TESTID;  
    RETURN NULL;  
END;  
'LANGUAGE plpgsql;
```

```
CREATE TRIGGER UPDATE_FORMS_NUM_DOWN AFTER DELETE  
ON FORM FOR EACH ROW  
EXECUTE PROCEDURE UPDATE_FORMS_NUMDOWN();
```




PLPGSQL (TRIGGERS)

- EJEMPLO:

```
CREATE FUNCTION NEW_NEWS_HISTORY()  
RETURNS TRIGGER AS'  
DECLARE  
    LASTNUM INT;  
BEGIN  
    LOCK TABLE NEWSHISTORY IN EXCLUSIVE MODE;  
    SELECT INTO LASTNUM MAX(MODNUM) FROM NEWSHISTORY WHERE  
NEWSID=OLD.NEWSID;  
    IF (LASTNUM IS NULL) THEN  
        LASTNUM:=0;  
    END IF;  
    LASTNUM:=LASTNUM+1;  
    INSERT INTO NEWSHISTORY (MODNUM,NEWSID,NEWSOWNER,INSERTDATE,NEWS)  
VALUES (LASTNUM,OLD.NEWSID,OLD.NEWSOWNER,OLD.LASTMOD,OLD.NEWS);  
    RETURN NEW;  
END;  
'LANGUAGE plpgsql;  
  
CREATE TRIGGER NEWS_RECORDER BEFORE UPDATE  
ON NEWS FOR EACH ROW EXECUTE PROCEDURE NEW_NEWS_HISTORY();
```


BLOQUES

Definición

En PostgreSQL, un bloque es una sección de código que agrupa una serie de declaraciones y sentencias que se ejecutan de forma secuencial. Los bloques son una característica fundamental del lenguaje procedimental PL/pgSQL, que es un lenguaje que permite escribir procedimientos almacenados, funciones, y triggers con estructuras de control de flujo (como bucles y condiciones) más avanzadas que las que permite el SQL estándar.

¿Por qué utilizar bloques?

- Modularidad: Permiten agrupar lógica compleja y fragmentarla en secciones manejables.
- Reutilización: El código dentro de un bloque puede ser reutilizado en múltiples lugares, especialmente cuando se usa dentro de funciones o procedimientos.
- Control de Flujo Avanzado: Los bloques permiten el uso de estructuras de control como bucles (FOR, WHILE) y condiciones (IF, CASE), que no son posibles en el SQL estándar.
- Manejo de Excepciones: Puedes manejar errores de manera sofisticada, asegurando que tu aplicación o consulta pueda responder adecuadamente a los problemas sin fallar abruptamente.



¿Qué es un
bloque es
PostgreSQL?

```
DO $$ -- Inicia un bloque anónimo de código PL/pgSQL.
DECLARE
    r RECORD; -- Declara una variable de tipo RECORD para almacenar los resultados de la consulta.
BEGIN
    FOR r IN (SELECT tablename FROM pg_tables WHERE schemaname = 'public') LOOP -- Itera sobre cada tabla en el esquema 'pub
        EXECUTE format('
            CREATE TRIGGER trg_auditoria_insert_%I
            AFTER INSERT ON %I
            FOR EACH ROW EXECUTE FUNCTION auditoria_insert();', r.tablename, r.tablename); -- Crea un trigger de INSERT para

        EXECUTE format('
            CREATE TRIGGER trg_auditoria_delete_%I
            AFTER DELETE ON %I
            FOR EACH ROW EXECUTE FUNCTION auditoria_delete();', r.tablename, r.tablename); -- Crea un trigger de DELETE para

        EXECUTE format('
            CREATE TRIGGER trg_auditoria_update_%I
            AFTER UPDATE ON %I
            FOR EACH ROW EXECUTE FUNCTION auditoria_update();', r.tablename, r.tablename); -- Crea un trigger de UPDATE para
        END LOOP; -- Finaliza el bucle FOR.
    END $$; -- Termina el bloque de código PL/pgSQL.
```


TAREA 2



Crear una base de datos “Tarea2”. Luego crear dos tablas “audit” y “usuario” (los campos quedan a discreción).

Configurar triggers (update, delete, insert) en “usuario” y guardar bitácora de operaciones en audit.

<https://josejuansanchez.org/bd/ejercicios-consultas-sql/>

Ejercicios. Realización de consultas SQL

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2023/2024

Índice

- 1 Ejercicios. Realización de consultas SQL
 - 1.1 Tienda de informática
 - 1.1.1 Modelo entidad/relación
 - 1.1.2 Base de datos para MySQL
 - 1.1.3 Consultas sobre una tabla
 - 1.1.4 Consultas multitabla (Composición interna)
 - 1.1.5 Consultas multitabla (Composición externa)
 - 1.1.6 Consultas resumen
 - 1.1.7 Subconsultas (En la cláusula WHERE)
 - 1.1.7.1 Con operadores básicos de comparación
 - 1.1.7.2 Subconsultas con ALL y ANY
 - 1.1.7.3 Subconsultas con IN y NOT IN

RECOMENDACION DE LA SEMANA