Prueba técnica Conexia – Desarrollaor java

Jose Alejandro Rivera Reyes

C.C. 1.022.416.139

**GENERALIDADES**

Conexia S.A.S es una empresa que tiene como función realizar software para el sector de la salud. Cada una de las soluciones implementadas en la organización toman como base el estandar de Java EE.

**CONSIDERACIONES**

Para resolver la prueba, el candidato debe leer y analizar completa y detenidamente el caso de estudio planteado; los entregables son:

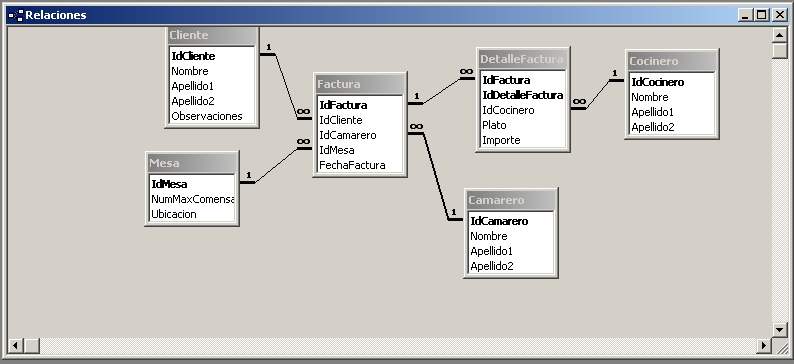
* Script de BD especificando el motor de BD en el que se trabaja.
* Aplicación web con aplicación (war, ear, etc)
* Codigo fuente de la aplicación que compile y permita realizar su ejecución.
* README con los datos necesarios para realizar el despliegue de la aplicación y uso de la misma.

El candidato para esta prueba necesita conocer:

* Manejo de gestores de BD (Mysql, Oracle, Postgresql, etc)
* Manejo y creación de aplicaciones web en Java
* Desarrollo de aplicaciones basadas en el principio de programación IoC
* Tener en cuenta la implementacion de patrones de diseño
* Desarrollo de aplicación en n-capas.
* Manejo en herramientas de gestion y construccion de proyectos (maven, gradle, etc).
* Implementación en frameworks de persistencia.

**CASO DE ESTUDIO**

Realizar la creación de este modelo E/R en BD para realizar la implementación a evaluar.



Se desea crear una aplicación que permita registrar las facturas de las ventas realizadas en el restaurante “LA MEJOR COCINA”. La aplicación debe contener el siguiente listado de requerimientos.

1. Registro de la factura donde se diligencie el cliente y el camarero que atendió la mesa. Adicionalmente debe permitir diligenciar los platos consumidos por el cliente.
2. Crear una funcionalidad que permita visualizar la siguiente información (No reporteador):
   * Mostrar nombre, apellido, mes y la suma del total facturado por cada camarero por mes; tenga en cuenta que deben verse inclusive los camareros que no facturaron nada.
   * Construya una consulta que muestre los clientes con más de cien mil pesos gastados en el restaurante

Respuesta:

De acuerdo a lo anteriormente solicitado envió los scripts a ejecutar en la base de datos “PostgresSql”.

-- **Usuario**: jrivera

CREATE USER jrivera WITH

LOGIN

SUPERUSER

CREATEDB

CREATEROLE

INHERIT

NOREPLICATION

CONNECTION LIMIT -1

PASSWORD 'jrivera';

* **creación de la base de datos**

CREATE DATABASE restaurant

WITH

OWNER = jrivera

ENCODING = 'UTF8'

CONNECTION LIMIT = -1;

* **creación del schema**

CREATE SCHEMA restaurant

AUTHORIZATION jrivera;

* **Creacion de la tabla cliente**

-- Table: restaurant.cliente

-- DROP TABLE restaurant.cliente;

CREATE TABLE restaurant.cliente

(

"idCliente" integer NOT NULL DEFAULT nextval('restaurant."cliente\_idCliente\_seq"'::regclass),

nombre character varying(60) COLLATE pg\_catalog."default" NOT NULL,

apellido1 character varying(60) COLLATE pg\_catalog."default" NOT NULL,

apellido2 character varying(60) COLLATE pg\_catalog."default",

observaciones character varying(100) COLLATE pg\_catalog."default",

CONSTRAINT pkcliente PRIMARY KEY ("idCliente")

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.cliente

OWNER to jrivera;

* **Creacion de la Tabla Mesa**

-- DROP TABLE restaurant.mesa;

CREATE TABLE restaurant.mesa

(

"idMesa" integer NOT NULL DEFAULT nextval('restaurant."mesa\_idMesa\_seq"'::regclass),

"numMaxComensales" numeric(3,0) NOT NULL,

ubicacion character varying(20) COLLATE pg\_catalog."default",

CONSTRAINT "pkMesa" PRIMARY KEY ("idMesa")

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.mesa

OWNER to jrivera;

* **Creacion de la tabla Cocinero**

-- DROP TABLE restaurant.cocinero;

CREATE TABLE restaurant.cocinero

(

"idCocinero" integer NOT NULL DEFAULT nextval('restaurant."cocinero\_idCocinero\_seq"'::regclass),

nombre character varying(30) COLLATE pg\_catalog."default" NOT NULL,

apellido1 character varying(30) COLLATE pg\_catalog."default" NOT NULL,

apellido2 character varying(30) COLLATE pg\_catalog."default",

CONSTRAINT pkcocinero PRIMARY KEY ("idCocinero")

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.cocinero

OWNER to jrivera;

OWNER to jrivera;

* **Creacion de la tabla Camarero**

-- DROP TABLE restaurant.camarero;

CREATE TABLE restaurant.camarero

(

"idCamarero" integer NOT NULL DEFAULT nextval('restaurant."camarero\_idCamarero\_seq"'::regclass),

nombre character varying(30) COLLATE pg\_catalog."default" NOT NULL,

apellido1 character varying(30) COLLATE pg\_catalog."default" NOT NULL,

apellido2 character varying(30) COLLATE pg\_catalog."default",

CONSTRAINT "pkCamarero" PRIMARY KEY ("idCamarero")

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.camarero

OWNER to jrivera;

* **Creacion de la tabla Detalle Factura**

-- DROP TABLE restaurant."detalleFactura";

CREATE TABLE restaurant."detalleFactura"

(

"idDetalleFactura" integer NOT NULL DEFAULT nextval('restaurant."detalleFactura\_idDetalleFactura\_seq"'::regclass),

"idFactura" integer NOT NULL,

"idCocinero" integer NOT NULL,

plato character varying(50) COLLATE pg\_catalog."default" NOT NULL,

importe numeric NOT NULL,

CONSTRAINT "pkDetalleFactura" PRIMARY KEY ("idDetalleFactura"),

CONSTRAINT "pfCocinero" FOREIGN KEY ("idCocinero")

REFERENCES restaurant.cocinero ("idCocinero") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "pfFactura" FOREIGN KEY ("idFactura")

REFERENCES restaurant.factura ("idFactura") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."detalleFactura"

OWNER to jrivera;

* **Creacion de la tabla factura**

-- DROP TABLE restaurant.factura;

CREATE TABLE restaurant.factura

(

"idFactura" integer NOT NULL DEFAULT nextval('restaurant."factura\_idFactura\_seq"'::regclass),

"idCliente" integer NOT NULL,

"idCamarero" integer NOT NULL,

"idMesa" integer NOT NULL,

"fechaFactura" date NOT NULL,

CONSTRAINT "pkFactura" PRIMARY KEY ("idFactura"),

CONSTRAINT "pfCamarero" FOREIGN KEY ("idCamarero")

REFERENCES restaurant.camarero ("idCamarero") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "pfCliente" FOREIGN KEY ("idCliente")

REFERENCES restaurant.cliente ("idCliente") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "pfMesa" FOREIGN KEY ("idMesa")

REFERENCES restaurant.mesa ("idMesa") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.factura

OWNER to jrivera;

* **Creación de la tabla platos**

CREATE TABLE restaurant.platos

(

idplato integer NOT NULL DEFAULT nextval('restaurant.platos\_idplato\_seq'::regclass),

descripcion character varying(70) COLLATE pg\_catalog."default" NOT NULL,

valor numeric NOT NULL,

idcocinero integer,

CONSTRAINT platos\_pkey PRIMARY KEY (idplato),

CONSTRAINT pfplatos FOREIGN KEY (idcocinero)

REFERENCES restaurant.cocinero (idcocinero) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant.platos

OWNER to jrivera;

* **insertar datos en la tabla cocinero**

INSERT INTO restaurant.cocinero(

nombre, apellido1, apellido2)

VALUES ('yerson', 'cortes', 'flores')

,('diego', 'camelo', 'serna')

,('camilo', 'ariza', 'cortina')

,('edwin', 'sambrano', 'rey');

* **Insertar datos en la tabla Platos**

INSERT INTO restaurant.platos(descripcion, valor,idcocinero)

VALUES ('mojarra frita', 12000,1)

,('arroz paisa', 12000,3)

,('Carne Frita',15000,1)

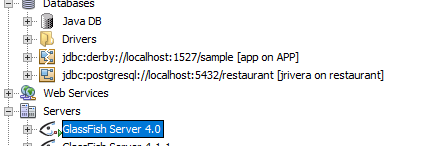
,('alitas BBQ',20000,2)

,('pollo con champillones', 13000,1)

,('costillitas BBQ',16000,3);

* **Configuracion del Data Source en el id de netbeans**

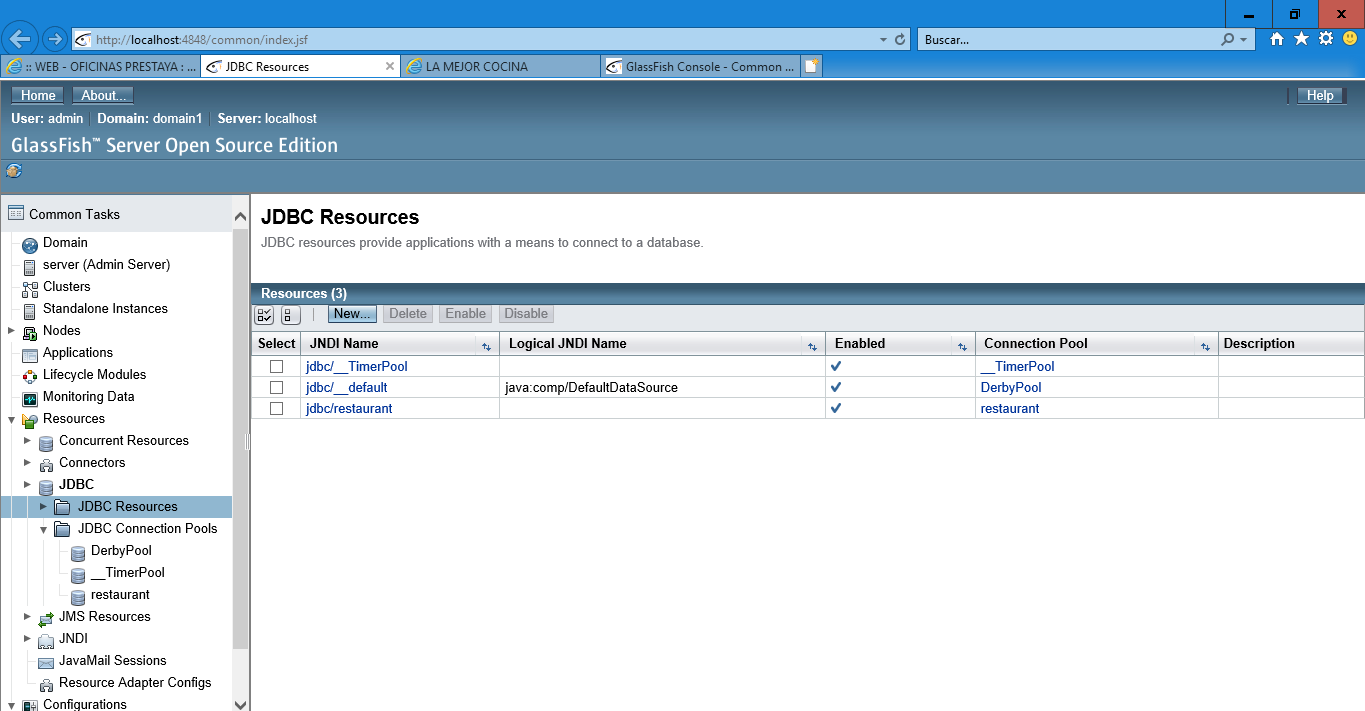
Ingresar en la opción **services**  y en la opción **Databases** crar la conexión de la aplicación a la base de datos para montar la persistencia en el servidor GlassFish4.0



Luego ingresar al servidor de glassFish para configurar el pool de conexiones.

Jdbc/restaurant[jdbc/restaurantjdbc/restaurant](http://localhost:4848/jdbc/jdbcResourceEdit.jsf?name=jdbc/restaurant&logicalJndiName=)

<http://localhost:4848/common/index.jsf>



Luego desplegar la aplicación WEB con el id de netbeans.