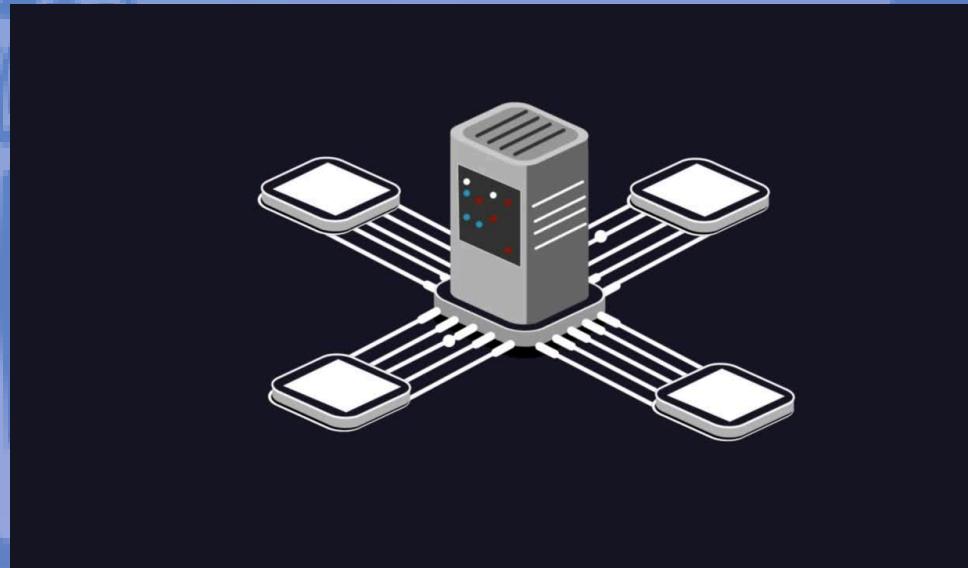


BASES DE DATOS



PROFESOR

Ing. Fernando Arreola

EQUIPO

UH?

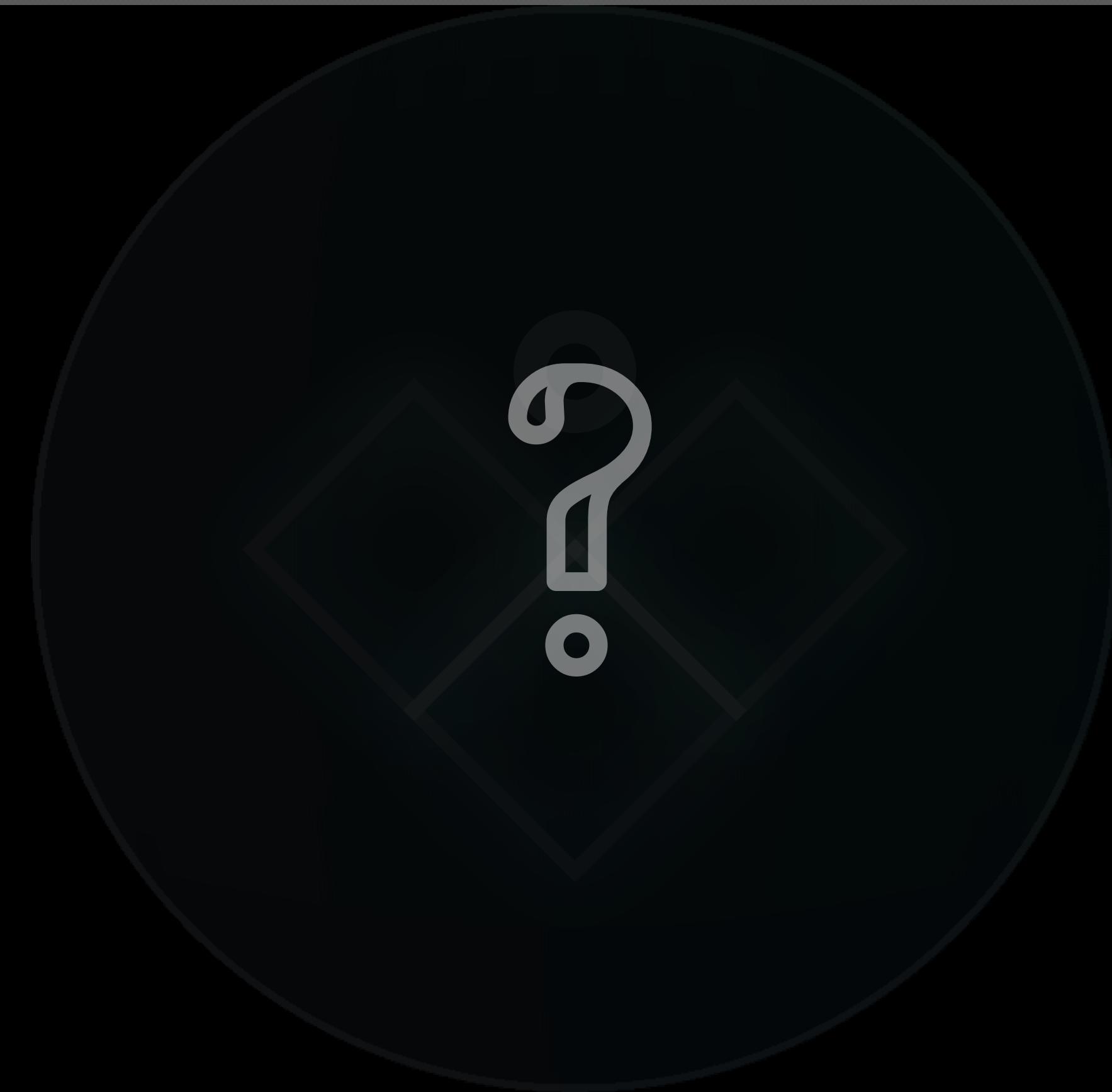
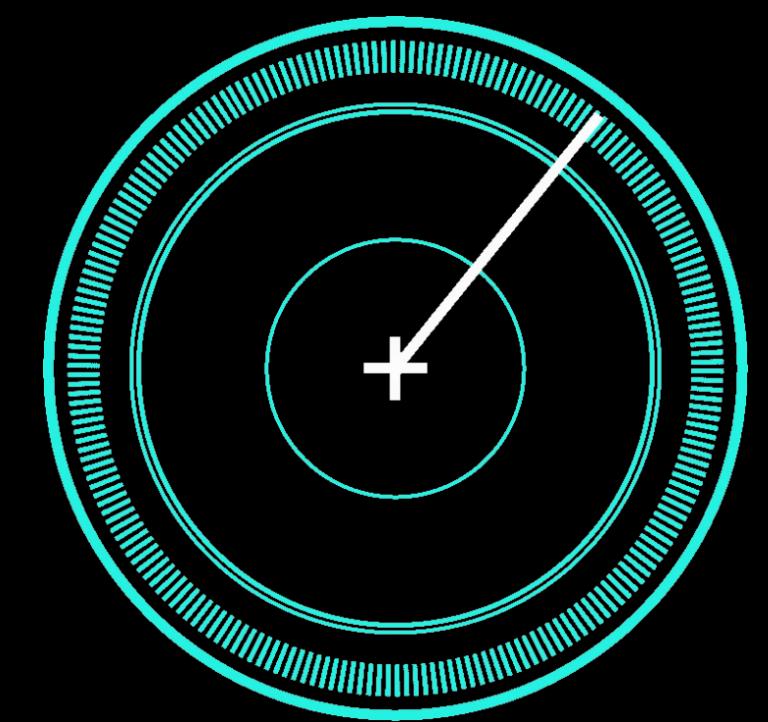
Hernandez Ramirez Miguel
Angel

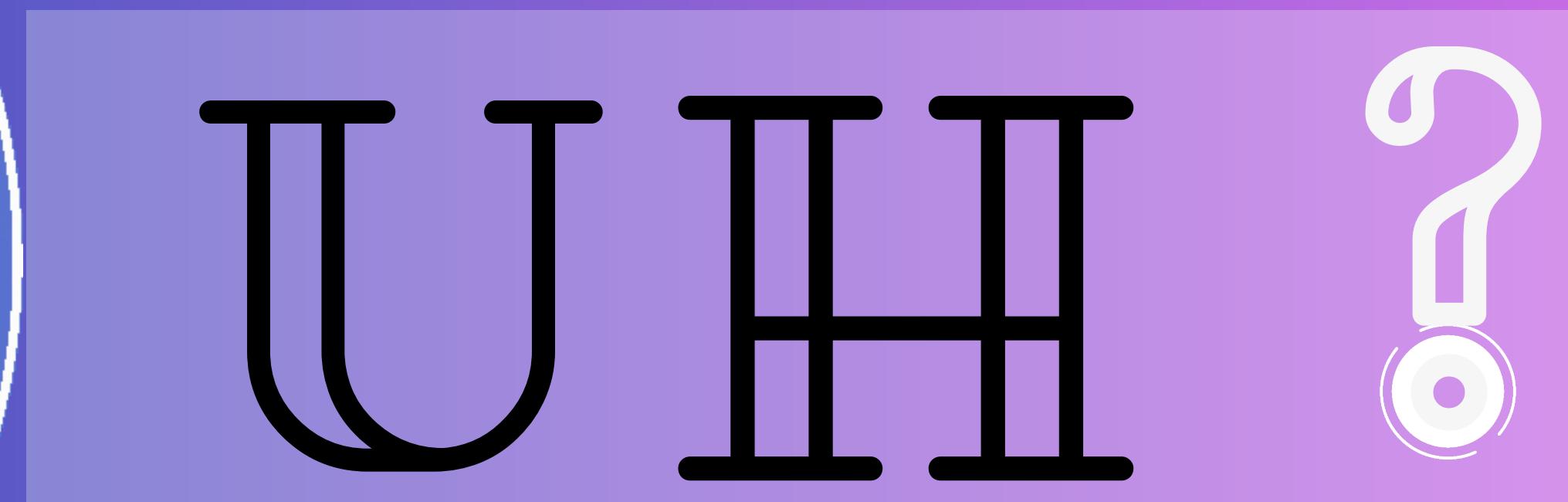
Rivera Lopez David Zaid

Soto Huerta Gustavo Isaac

Silverio Martinez Andres

Ugalde Santos Atzin

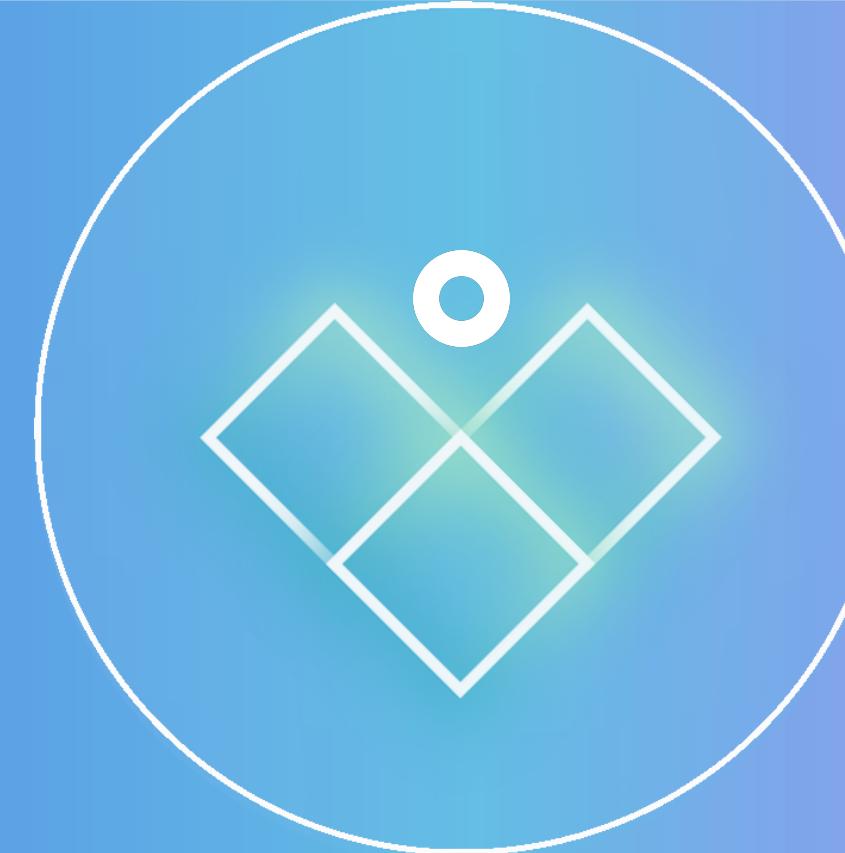


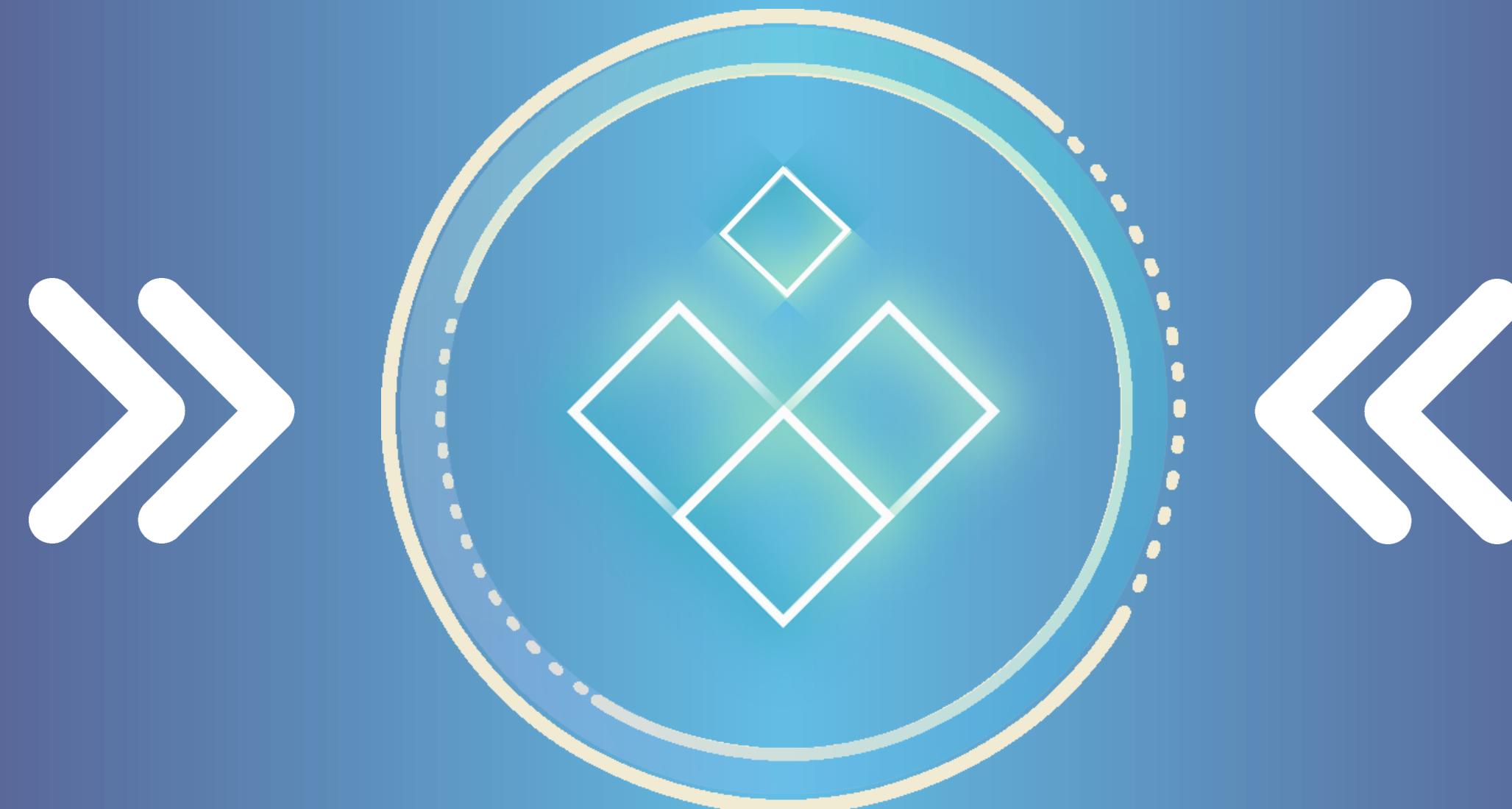


Universal

Harmony







TEAM



CONNECTING ALL OF US WITH THE WORLD



PROJECT MANAGER



DATABASE DESIGNER



FULL STACK DEVELOPER

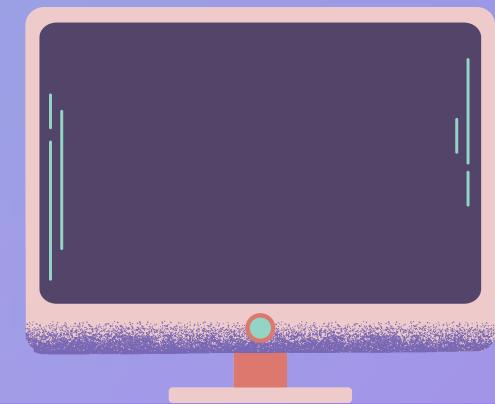
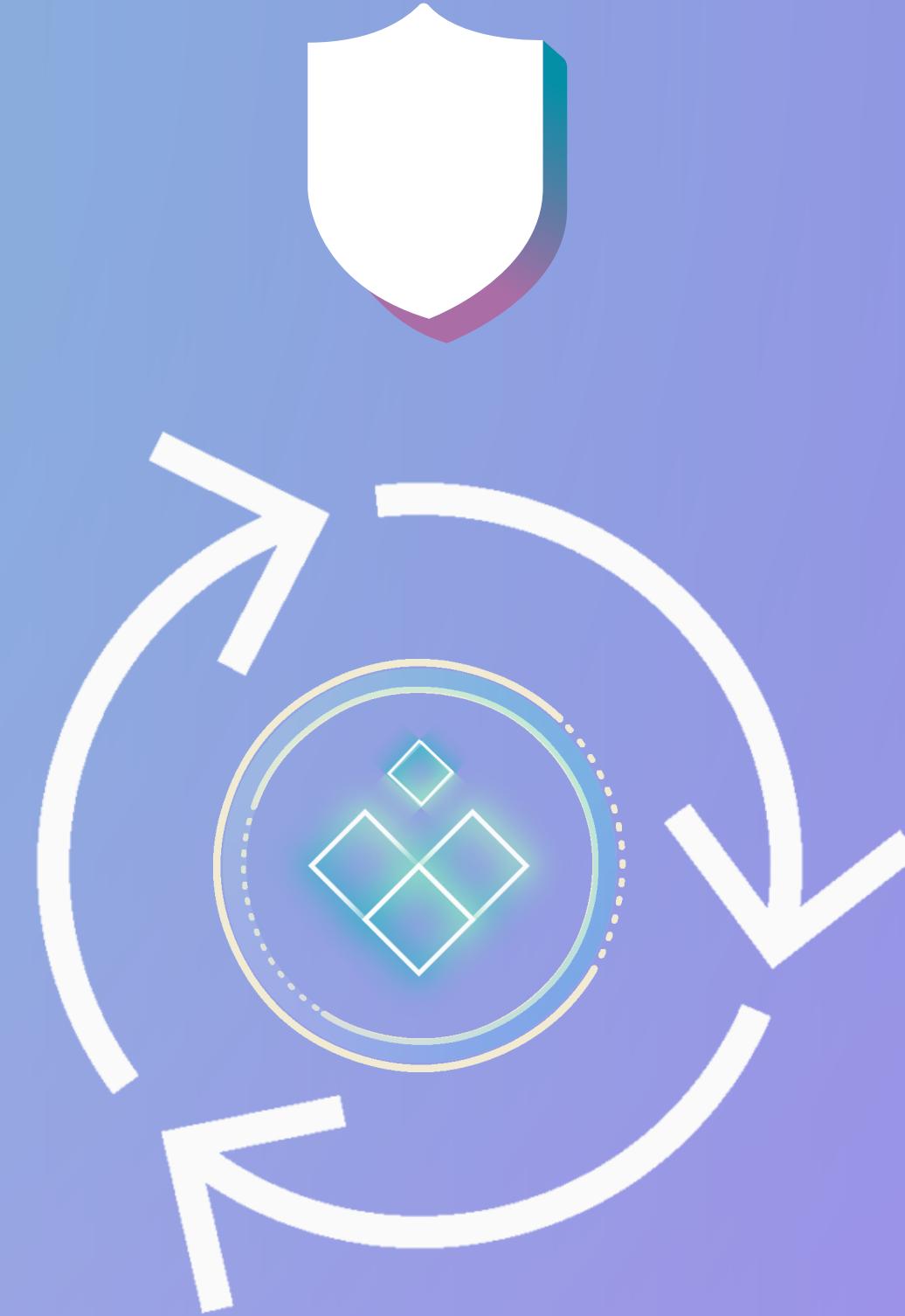


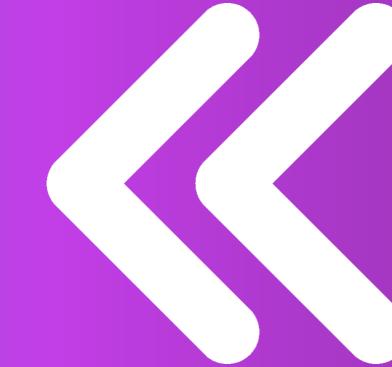
UI DESIGNER



TESTER





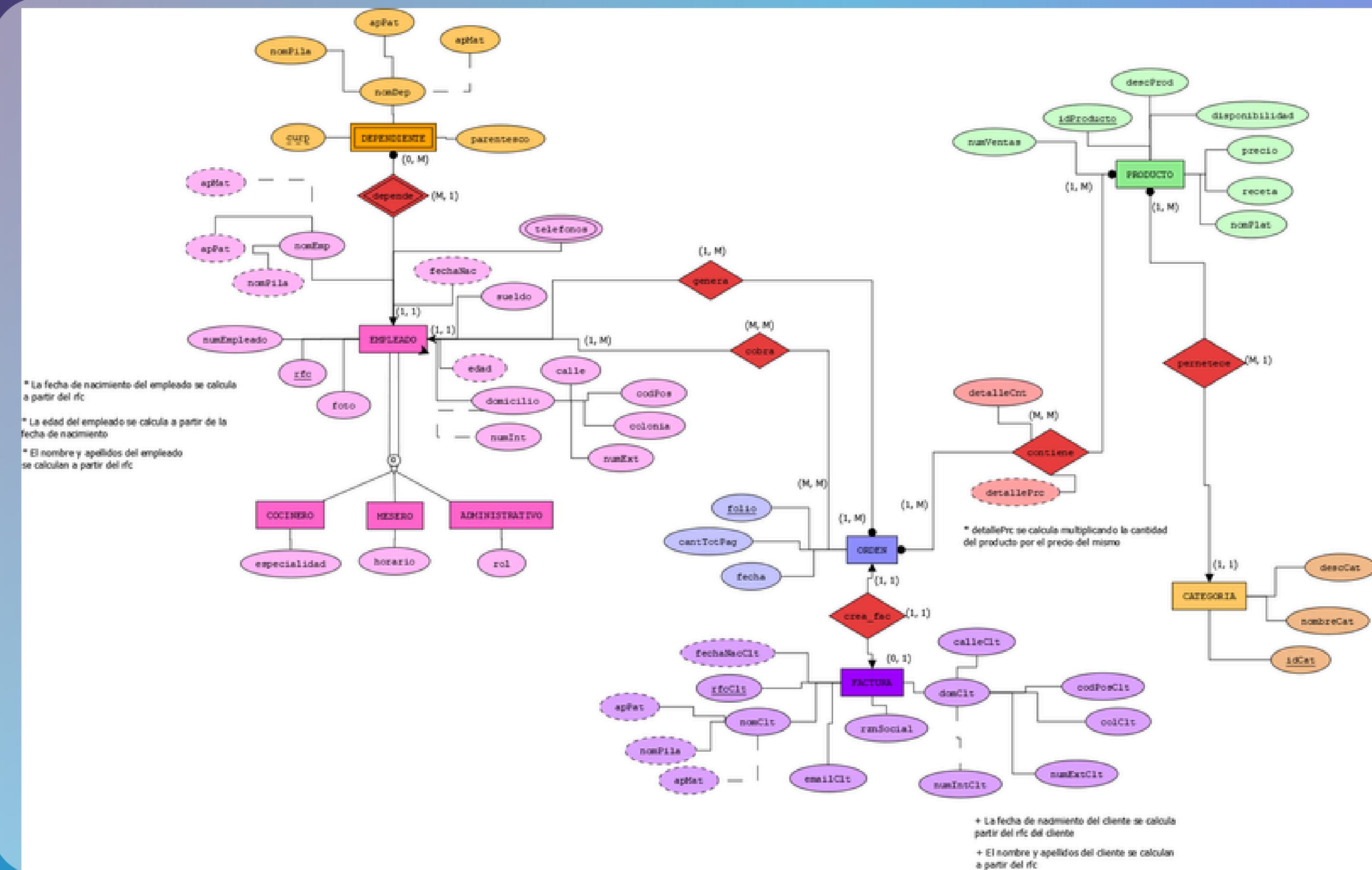


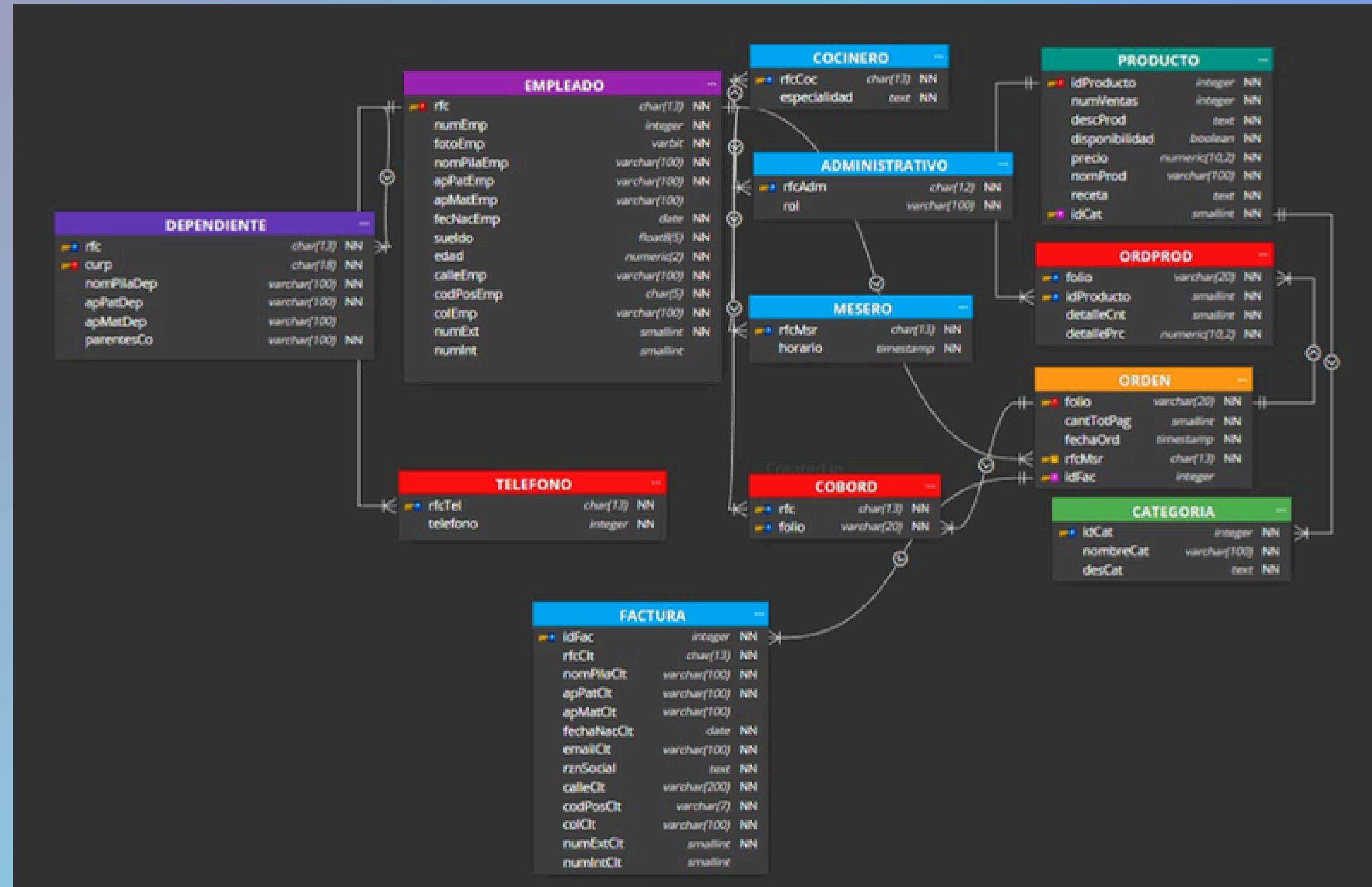
DEVELOPMENT

CONNECTING ALL OF US WITH THE WORLD

ANALISIS



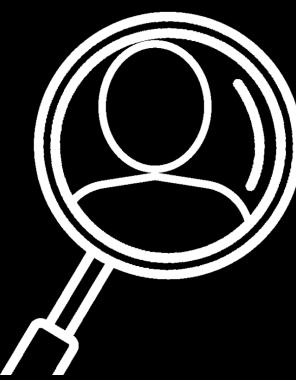






FINAL PROJECT DEMO

CONNECTING ALL OF US WITH THE WORLD



```
-- Calculando el total de prod en ORDPROD
CREATE OR REPLACE FUNCTION calculate_order_product_total() RETURNS TRIGGER AS $$
DECLARE
    product_price NUMERIC(10, 2);
BEGIN
    SELECT precio INTO product_price FROM "PRODUCTO" WHERE "idProducto" = NEW."idProducto"
    NEW."detallePrc" := NEW."detalleCnt" * product_price;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

REQUERIMIENTOS

PRIMER TESTEO



ACTUALIZACION DE ESTADO DE PRODUCTO



```
CREATE TRIGGER update_order_product_total  
BEFORE INSERT OR UPDATE ON "ORDPROD"  
FOR EACH ROW  
EXECUTE FUNCTION calculate_order_product_total();
```



REQUERIMIENTOS
SEGUNDO TESTEO

ACTUALIZACION DE VENTA



```
CREATE OR REPLACE FUNCTION check_product_availability() RETURNS TRIGGER AS $$  
BEGIN  
    IF NOT EXISTS (  
        SELECT 1 FROM "PRODUCTO"  
        WHERE "idProducto" = NEW."idProducto" AND disponibilidad = TRUE  
    ) THEN  
        RAISE EXCEPTION 'El producto no está disponible.';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER check_availability_before_insert_or_update  
BEFORE INSERT OR UPDATE ON "ORDPROD"  
FOR EACH ROW  
EXECUTE FUNCTION check_product_availability();
```

REQUERIMIENTOS
TERCER TESTEO



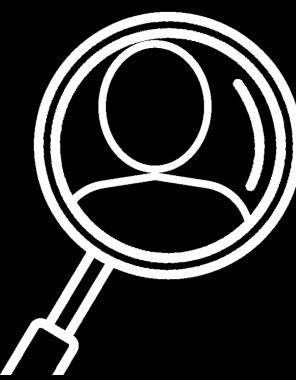
VALIDACION DE DISPONIBILIDAD DE PRODUCTO



REQUERIMIENTOS
CUARTO TESTEO



VISTA FACTURA



REQUERIMIENTOS
QUINTO TESTEO

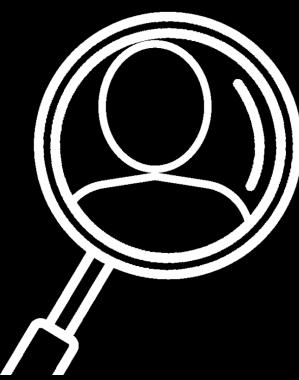
```
CREATE OR REPLACE FUNCTION empMesero(numEmp INTEGER, OUT ordenesAtendidas INTEGER, OUT totalPagado NUMERIC)
RETURNS RECORD AS
$$
DECLARE
    esMesero BOOLEAN;
BEGIN
    -- Verificar si el empleado es un mesero
    SELECT EXISTS(SELECT 1 FROM "MESERO" WHERE "rfcMsr" = (SELECT rfc FROM "EMPLEADO" WHERE "numEmp" = numEmp)) INTO esMesero;

    IF esMesero THEN
        -- Contar la cantidad de órdenes registradas por el mesero en el día
        SELECT COUNT(*), COALESCE(SUM("cantTotPag"), 0)
        INTO ordenesAtendidas, totalPagado
        FROM "ORDEN"
        WHERE "rfcMsr" = (SELECT rfc FROM "EMPLEADO" WHERE "numEmp" = numEmp);

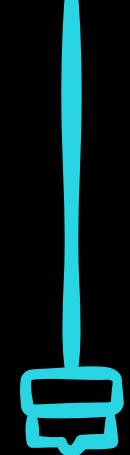
    ELSE
        -- Mostrar un mensaje de error si no es un mesero
        RAISE EXCEPTION 'El numero de empleado introducido no corresponde a un mesero';
    END IF;
END;
$$
LANGUAGE plpgsql;
```



VENTAS POR ORDENES



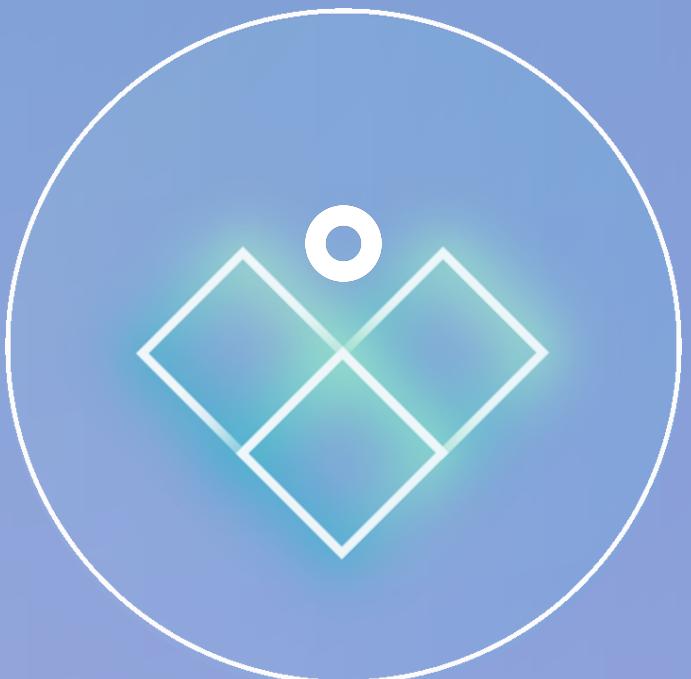
```
CREATE VIEW vistaProdMasVendido AS  
SELECT * FROM "PRODUCTO"  
WHERE "numVentas" = (SELECT MAX("numVentas") FROM "PRODUCTO");
```



REQUERIMIENTOS
SEXTO TESTEO

PRODUCTO MAS VENDIDO

AGRADECIMIENTO



APRECIAMOS SINCERAMENTE SU CONFIANZA EN LA EMPRESA UH

