

Universidad Autónoma de Querétaro  
Facultad de Informática



## BASE DE DATOS I

**Carrera:**

Ingeniería de Software.

**Profesora:** M.S.C. Sofía Amadís Rivera López

**Alumno:** Cano Cabrera David Emmanuel

## PROYECTO FINAL

**Sistema de Información** para resolver  
las necesidades de la empresa **Dave's**

06 de diciembre de 2021

## Contenidos

<b>Introducción</b> .....	2
<b>Análisis de requerimientos</b> .....	3
<b>Planteamiento del Problema</b> .....	3
<b>Necesidad</b> .....	4
<b>Justificación de desarrollo del sistema</b> .....	4
<b>Propuesta de solución:</b> .....	4
<b>Tecnologías para emplear</b> .....	5
<b>Propuesta de valor</b> .....	5
<b>Funcionalidades y características</b> .....	6
<b>Perspectiva del producto</b> .....	6
<b>Restricciones</b> .....	7
<b>Suposiciones y dependencias</b> .....	7
<b>Requerimientos específicos</b> .....	8
<b>Requerimientos comunes de las interfaces</b> .....	8
<b>Requerimientos funcionales (procesos)</b> .....	8
<b>Requerimientos no funcionales</b> .....	10
<b>Diseño de la base de datos</b> .....	11
<b>Entidades</b> .....	11
<b>Atributos</b> .....	11
<b>Diagrama entidad relación</b> .....	13
<b>Diagrama entidad relación con cardinalidad y atributos</b> .....	14
<b>Modelo físico de datos</b> .....	15
<b>Sentencias SQL</b> .....	16
<b>Análisis del sistema</b> .....	19
<b>Justificación de la metodología de desarrollo</b> .....	19
<b>Diseño del sistema</b> .....	21
<b>Bibliografía</b> .....	22

## Introducción

Resiliencia: capacidad para sobreponerse ante una situación adversa e incluso, salir fortalecido de ella. Una definición muy astuta utilizada hoy en día para expresar el proceso de los actuales comercios, comunidades y sociedades en general para reconstruirse en la búsqueda de sobrellevar los problemas que la crisis de salud mundial supuso para todos. Y qué más allá de representar la constante lucha interna de los individuos, simboliza un esfuerzo colectivo por salir adelante. De donde sus resultados, han sido galardonados para todos aquellos que se han adaptado correctamente al cambio.

Tal es el caso, de la empresa “Dave’s”: una lonchería que durante la pandemia ha conseguido modificar su modelo de negocios en pro de mantenerse a flote. Adaptando un estilo de venta de puerta en puerta, obteniendo rédito a partir de un equipo de repartidores (distribuidores del producto) que de una manera muy informal aseguran una relación con los clientes consumidores de los insumos alimenticios que la empresa Dave’s ofrece para ellos.

Esta situación ha sacado a la luz una infinidad de áreas de oportunidad, mismas que desde un enfoque tecnológico serán abordadas a través de este documento. Y cuyo objetivo principal será dar a conocer el proceso en el modelado de un sistema de información que se adapte a las necesidades de la actual empresa, permitiendo maximizar procesos, automatizar tareas y desde un punto de vista estratégico como nos indica Briceño (2013), “proporcionar una ventaja frente a los demás competidores dentro del mercado”.

A través de la identificación de una problemática, la generación de una propuesta, su justificación y puesta en marcha. Abordando las etapas de diseño y análisis del sistema. El cual, en su conjunto, establecerá las pautas para llevar a cabo su implementación en un entorno productivo.

## **Análisis de requerimientos**

### **Planteamiento del Problema**

Actualmente, la empresa “Dave’s” utiliza una cocina a puertas cerradas donde se encarga de preparar una variedad de productos para su venta individual, utilizando a “repartidores” como distribuidores de estos, el dueño se encarga de dividir las áreas de la ciudad en dónde cada repartidor va a operar (las cuales son llamadas rutas) y según la experiencia del repartidor se le asignará una cuota de venta mínima la cual deberá de ser cumplida para antes de terminar sus horas de servicio. Al finalizar la jornada el repartidor debe personalmente establecer el número de piezas mínimas que puede vender, sin embargo, durante su jornada muchas veces esta cuota o no es cumplida o no se cuenta con las suficientes piezas para cubrir su zona por completo. Esto obliga a llamar a la cocina para la producción de más piezas, lo que ocasiona atrasos en los pedidos realizados a “la carta”. Los pedidos a “la carta” forman parte del menú público que se encuentra disponible a los clientes, esto también significa que los clientes no pueden saber cuántas piezas “de ruta” se encuentran en existencia. logísticamente este problema se intenta resolver utilizando una red social para indicar a cocina las piezas que un repartidor necesita, específicamente un grupo de WhatsApp, este mismo grupo es utilizado en el momento en que un repartidor necesita consultar las piezas de un compañero buscando el intercambio y reasignación de piezas para poder cubrir los clientes de su ruta.

### **Necesidad**

El constante llamado a cocina y el uso de una única red social para gestionar la *última milla*, entorpece la organización de los encargados de producción, cuyo problema afecta al día a día de la empresa, lo que vuelve extremadamente relevante la acción para mitigarlo.

El objetivo de este proyecto es generar un sistema de información que trabaje alrededor de la lógica de negocio existente, priorizando de mejor manera las necesidades de nuestros clientes y actuando de acuerdo con la estrategia de venta

de cada repartidor, permitiéndole ser más flexible a la hora de asignar su cuota mínima de venta.

### **Justificación de desarrollo del sistema**

#### **Propuesta de solución:**

A partir del análisis de la problemática y los diversos actores que participan en ella. Se ha advertido como necesario un sistema de información especializado, que permita a los clientes llevar a cabo la solicitud de pedidos a domicilio utilizando una aplicación móvil basada en Android (identificado como el sistema operativo con mayor disponibilidad en el mercado). Arquitectura que se repetirá para la estación que se encargará de recibir esos pedidos. El sistema a partir del registro de las actividades dentro de una base de datos se encargará de mantener el sistema disponible durante las horas de ejecución del servicio de venta a domicilio de la empresa.

#### **Tecnologías para emplear**

El almacenamiento persistente de los datos será llevado a cabo utilizando el motor de BD **Oracle Database Estándar Edition**, a través del soporte técnico integral de Microsoft y Oracle utilizando **Oracle Cloud Infraestructure con Microsoft Azure**. Esto asegurará el soporte y disponibilidad necesario del sistema de acuerdo con las exigencias del entorno de producción. Su diseño y prueba será llevado a cabo de manera local utilizando como herramienta visual **Oracle SQL Developer**.

Para el maquetado de la solución se utilizará el software **Figma**, En el cual se creará el sistema de diseño ajustado a las heurísticas actuales.

En la fase de codificación del proyecto se espera el uso de tecnologías nativas de cada sistema operativo móvil para el despliegue de la solución, **Java** o Kotlin para Android y **Swift** para iOS. Priorizando el desarrollo Android. Estableciendo como evaluada la opción de utilizar una herramienta para el desarrollo de sistema multiplataforma basada en el lenguaje Dart como Flutter como lo indica Diví (2020).

### Propuesta de valor

Es necesario reconocer que existen actualmente en el mercado, soluciones que permiten a través de herramientas digitales, gestionar el proceso de venta de productos alimenticios, sin embargo, muchos de estos servicios no se encuentran disponibles en la región dónde se encuentran operando la empresa Dave's.

Además, el modelo de negocios y la alta competitividad, demanda una solución in-house que permita destacar frente a los demás competidores. Misma solución agrega la posibilidad de gestionar de manera más efectiva la solicitud de productos catalogados por la empresa lo que volvería más eficiente el trámite de las necesidades del cliente para con la misma. Cosa que no se podría lograr de no incrustar la lógica de negocio actual a una aplicación o funcionalidad creada de manera personalizada.

### Funcionalidades y características

#### Perspectiva del producto

El sistema de información para "Dave's" intentará automatizar el proceso de solicitud de pedidos y gestión de estos, administrándolos en función de su prioridad y según el personal disponible, centralizando el acceso a información de los clientes, pedidos y repartidores asignados a su cargo.

#### Características de los usuarios

<b>Tipo de Usuario</b>	Administrador
<b>Formación</b>	Dueño o gerente de Dave's
<b>Actividades</b>	Monitoreo de pedidos realizados.

<b>Tipo de Usuario</b>	Operador
<b>Formación</b>	Bachillerato terminado
<b>Actividades</b>	Recepción de órdenes, creación de comandas físicas, asignación de repartidores, comunicación con el cliente.

<b>Tipo de Usuario</b>	Repartidor
<b>Formación</b>	Licencia de conducir A2, conocimiento en manejo de motocicletas de 150 CC.
<b>Actividades</b>	Distribución de productos de ruta, recepción y entrega de productos.

<b>Tipo de Usuario</b>	Cliente
<b>Formación</b>	Ninguna
<b>Actividades</b>	Creación de pedido, adición, eliminación, modificación, de productos al pedido.

### Restricciones

- El sistema deberá de operar sobre una arquitectura de red con conexión a internet.
- Los servicios del sistema serán instalados en una arquitectura cliente/servidor.
- Se espera el uso de Android 6 o superior para los dispositivos que implementen el sistema.
- El diseño del sistema será sencillo e independiente de los lenguajes de programación utilizados.
- Se espera una escalabilidad horizontal en el número de dispositivos conectados a los servicios del sistema.
- Debe existir una aplicación de usuario independiente a la aplicación utilizada para los roles de administración, repartidor y operador.
- El sistema solo permitirá el inicio de sesión de repartidor a los usuarios dados de alta desde el rol de administrador.
- En caso de que una operación falle, el sistema del lado del servidor debe proveer una alternativa para verificar su estado.

- Un usuario puede acceder a la aplicación sin haber antes iniciado sesión.

### **Suposiciones y dependencias**

- Se asume que existirá una conexión basada en el estándar **IEEE 802.16** o **IEEE 802.11** en todos los dispositivos que se conecten al sistema.
- Los dispositivos de hardware utilizados como parte del sistema cumplen con un mínimo de requisitos necesarios para su correcto funcionamiento

### **Requerimientos específicos**

#### **Requerimientos comunes de las interfaces**

- Interfaces de Hardware:
  - o Para el correcto funcionamiento del sistema en dispositivos móviles se recomienda procesador Quad Core 1,2 GHz, al menos 2Gb de memoria y un almacenamiento igual o superior a los 16 Gb.
- Interfaces de Software:
  - o la versión de Oracle que se ejecutará en el servidor será la 21c Estándar Edition.
  - o El sistema móvil se ejecutará en las versiones iguales o posteriores a la versión marshmallow 6.0.
- Interfaces de Comunicación:
  - o La transferencia de información inalámbrica utilizara los medios wifi y bluetooth y estándares de red móviles.

#### **Requerimientos funcionales (procesos)**

##### **1. Acceso a la aplicación**

- El sistema permitirá al cliente registrarse a la aplicación utilizando una interfaz gráfica.
- La validación de su cuenta se dará una vez se haya ingresado el número de verificación enviado a su correo electrónico.
- Una vez validado el cliente tendrá acceso a las demás funcionalidades del sistema.



- El usuario administrador no debería de registrarse, pues contara con credenciales específicas para acceder al panel de configuración del sistema.
- El usuario administrador tendrá acceso a la creación de nuevos perfiles para los repartidores.
- Una vez creados los perfiles de repartidor y cliente estos deberían poder actualizar su información desde el panel de configuración.
- La información referente a la contraseña de los usuarios del sistema debe de estar protegida visualmente permitiéndole alternar la visibilidad de la contraseña.

## **2. Solicitud de pedido**

- El cliente podrá acceder a 5 secciones diferentes dentro de la aplicación: el menú de inicio donde se ubicarán de manera ordenada las categorías existentes, junto con un bloque de platillos de sugerencia. El segundo apartado contendrá las ordenes actuales o pendientes que tiene por se entregadas representara el carro de compras, el tercer apartado contendrá un faro que marcará su ubicación para que los repartidores con ruta cerca de su domicilio brinden su servicio. El cuarto será un chat con el cual podrá conversar directamente con el personal de atención al cliente y el ultimo mostrará su perfil con el acceso a configuraciones de su cuenta.
- El cliente también será capaz de llevar a cabo una búsqueda utilizando una barra superior. El resultado de la búsqueda desplegara de manera organizada los productos parecidos a la búsqueda.
- El usuario será redirigido al carro de compras una vez haya seleccionado los productos que desea comprar.
- El usuario será capaz de agregar comentarios para cada ítem de su orden junto con alguna nota de su pedido.
- modificación de pedido. El sistema le permitirá al usuario realizar cambios a las ordenes de su pedido antes de ser enviado a producción.

**3. Gestión de pedidos por el repartidor**

- Búsqueda de pedido.
- La interfaz de repartidor le permitirá consultar los pedidos asignados a su cargo, así como la opción de marcar el estado de cada pedido.
- Se le mostrara un mapa con la ruta y los posibles clientes potenciales que hayan decidido activar su faro para ser atendidos con el servicio de comida puerta a puerta.
- El sistema le mandara una alerta cuando un nuevo pedido haya sido registrado en la base de datos.

**4. Administración de los insumos**

- La interfaz de administración permitirá actualizar las categorías y productos que se encuentran disponibles para su consumo.
- El sistema capturara la modificación de los datos según la configuración de información necesaria para poder ser actualizado.
- El administrador tendrá acceso a la creación de ofertas desde su interfaz.

**5. Acceso a la lista de pedidos**

- El operador tendrá acceso a la lista de pedidos entrantes con la información necesaria para conocer quien lo recogerá y hacia donde se dirige.
- La vista de la interfaz del operador se centrará en facilitar la navegación entre pedidos.
- Una vez haya sido entregado un pedido por el repartidor estos dejaran de aparecer en el dashboard del operador.

**Requerimientos no funcionales****Rendimiento**

- Para asegurar el correcto funcionamiento del almacenamiento dentro de la infraestructura del servidor será necesario contar con 15% de capacidad de almacenamiento libre ajustados al SLA de los servicios en la Nube.

**Seguridad**

- Los permisos de acceso al sistema solo podrán ser asignados o revocados por el administrador.
- El SLA permitirá el respaldo de los sistemas cada 24 horas.
- La comunicación hacia servicios externos se llevará a cabo con las debidas credenciales otorgadas por la API implementada.

### Disponibilidad

- El sistema debe de estar disponible el 99.98% del tiempo en que cualquier usuario intente acceder al sistema.
- La probabilidad de falla del sistema no podrá ser de más del 5%.

### Organizacionales

- Para llevar a cabo la gestión del proyecto será necesario implementar una metodología ágil basada en SCRUM y DevOps.
- El sistema debe diseñarse utilizando el producto de software FIGMA.
- El IDE de programación sugerido es Android Estudio.

## Diseño de la base de datos

### Entidades



**Atributos**

Empleado	
idEmpleado	INTEGER NOT NULL PK
nombreEmpleado	VARCHAR2(150)
direccion	VARCHAR2(255)
numeroContacto	NUMBER(10,0)
email	VARCHAR2(255)
constrasena	VARCHAR2(30)
fechaIngreso	TIMESTAMP

Cliente	
idCliente	INTEGER NOT NULL PK
nombreCliente	VARCHAR2(150)
direccion	VARCHAR2(255)
numeroContacto	NUMBER(10,0)
email	VARCHAR2(255)
codigoConfirmacion	NUMBER(6,0)
contrasena	VARCHAR2(30)
fechaIngreso	TIMESTAMP

Estado	
idEstado	INTEGER NOT NULL PK
nombreEstado	VARCHAR(20)

Oferta	
idOferta	INTEGER NOT NULL PK
fechaInicio	DATE N
fechaFinal	DATE N
horaInicio	TIMESTAMP N
horaFinal	TIMESTAMP N
precioOferta	NUMBER(12,2)

Pedido	
idPedido	INTEGER NOT NULL PK
idSucursal	INTEGER NOT NULL FK
idCliente	INTEGER NOT NULL FK
idEmpleado	INTEGER NOT NULL FK
tiempoEntrada	TIMESTAMP
tiempoEstimadoLlegada	TIMESTAMP
tiempoLlegada	TIMESTAMP N
direccionEnvio	VARCHAR(255)
precio	NUMBER(12,2)
descuento	NUMBER(12,2)
precioFinal	NUMBER(12,2)
comentario	VARCHAR(255) N

Orden	
idOrden	INTEGER NOT NULL PK
idPedido	INTEGER NOT NULL FK
idOferta	INTEGER N FK
idProducto	INTEGER N FK
cantidad	INTEGER
precioProducto	NUMBER (12,2)
precioTotal	NUMBER (12,2)
comentario	VARCHAR(255) N

Producto	
idProducto	INTEGER NOT NULL PK
idCategoria	INTEGER NOT NULL FK
nombreProducto	VARCHAR(100)
descripcion	VARCHAR(225)

Categoría	
idCategoria	INTEGER NOT NULL PK
nombreCategoria	VARCHAR (100)

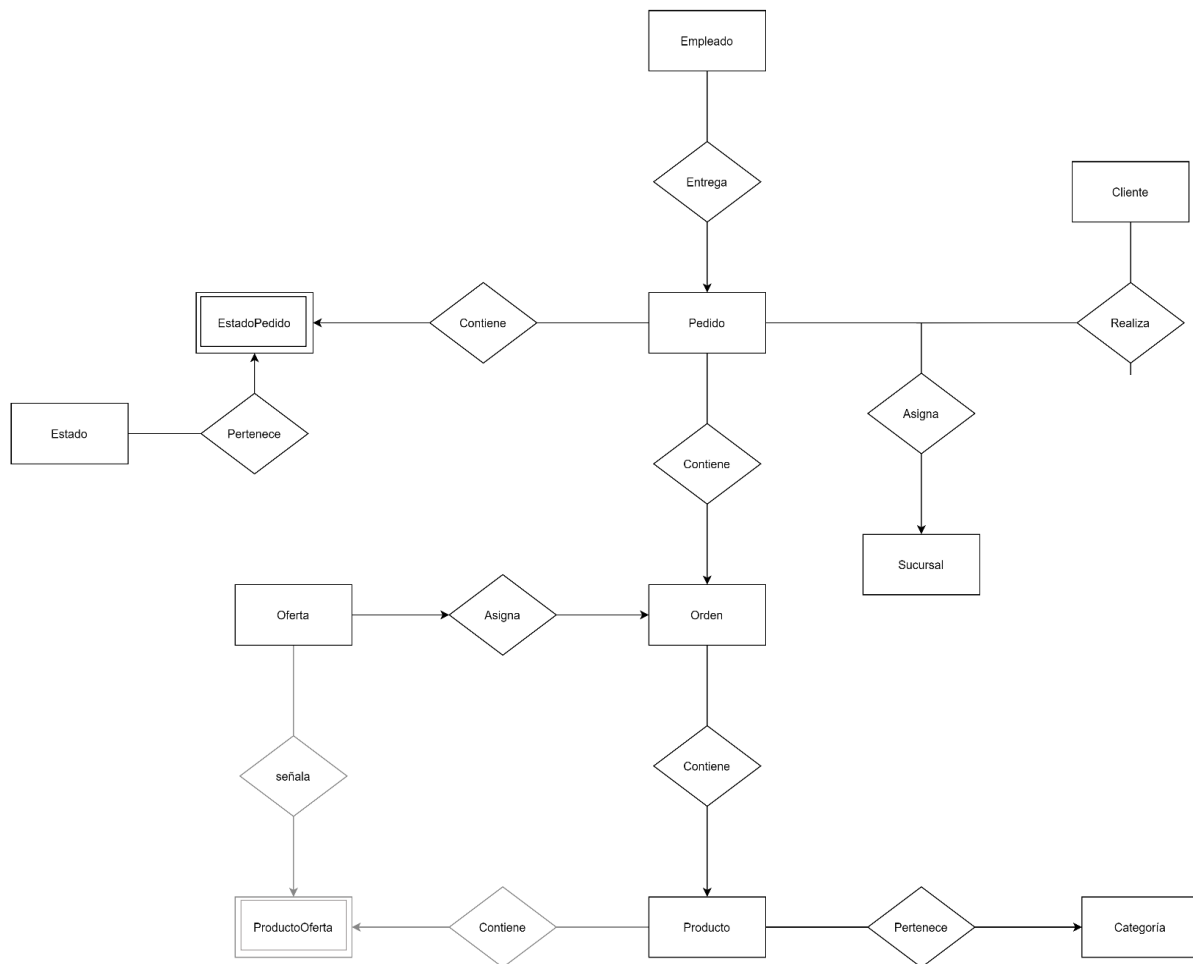
ingredientes	VARCHAR(500)
precio	NUMBER (12,2)
disponible	CHAR(1)

Sucursal	
idSucursal	INTEGER NOT NULL PK
direccion	VARCHAR(225)

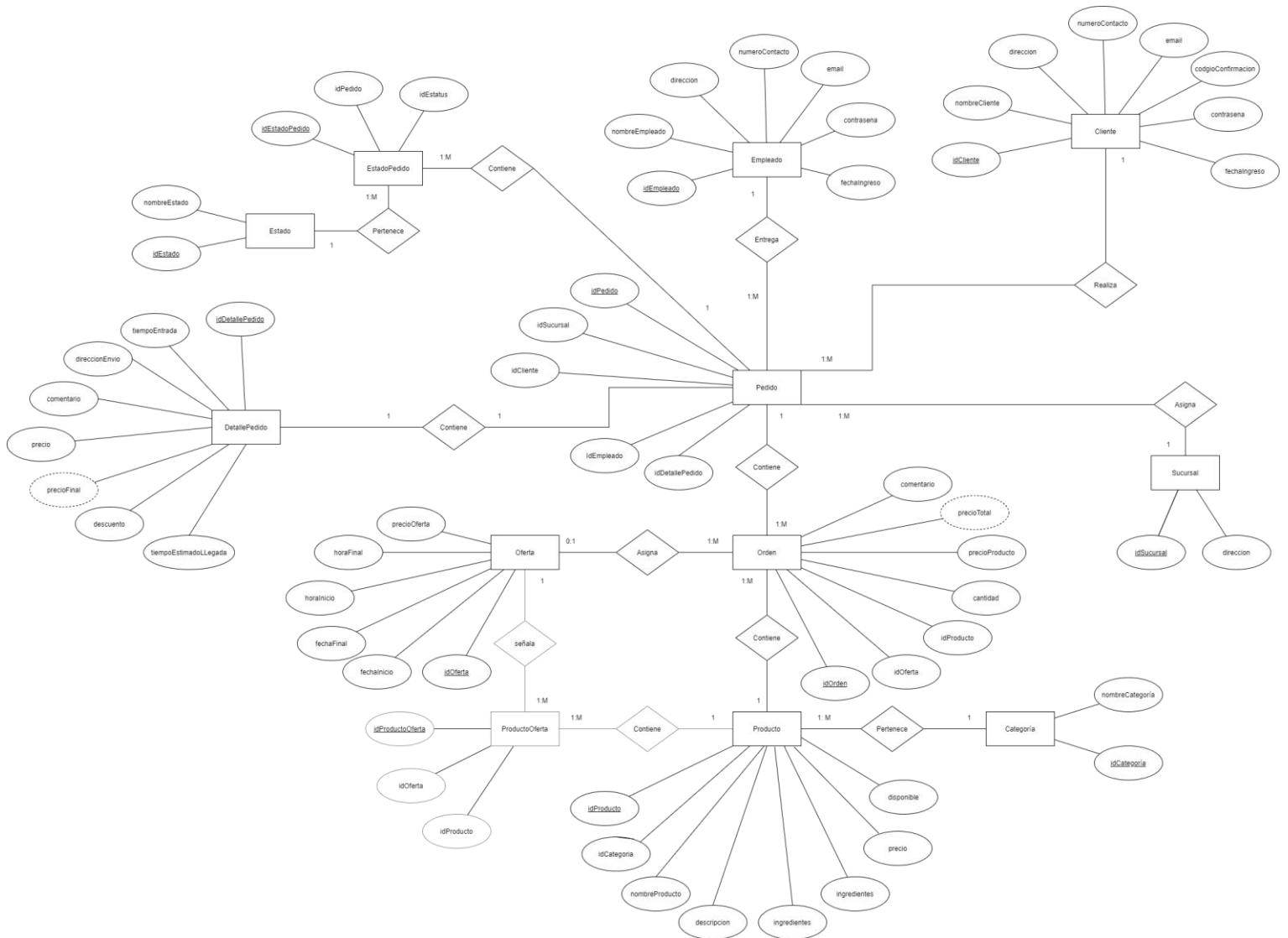
ProductoOferta	
idProductoOferta	INTEGER NOT NULL PK
idOferta	INTEGER NOT NULL FK
idProducto	INTEGER NOT NULL FK

EstadoPedido	
idEstadoPedido	INTEGER NOT NULL PK
idPedido	INTEGER NOT NULL FK
idEstatus	INTEGER NOT NULL FK

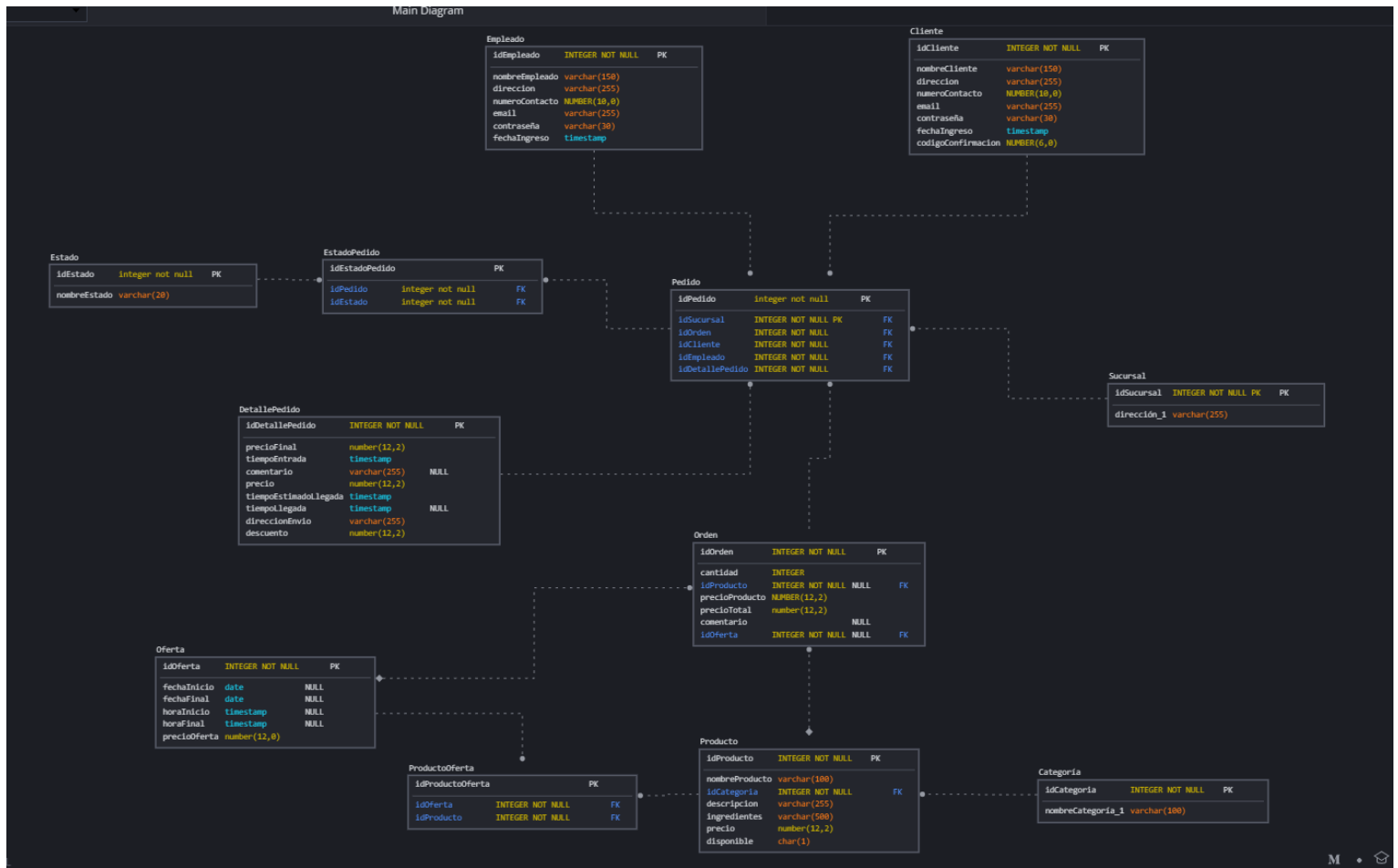
### Diagrama entidad relación



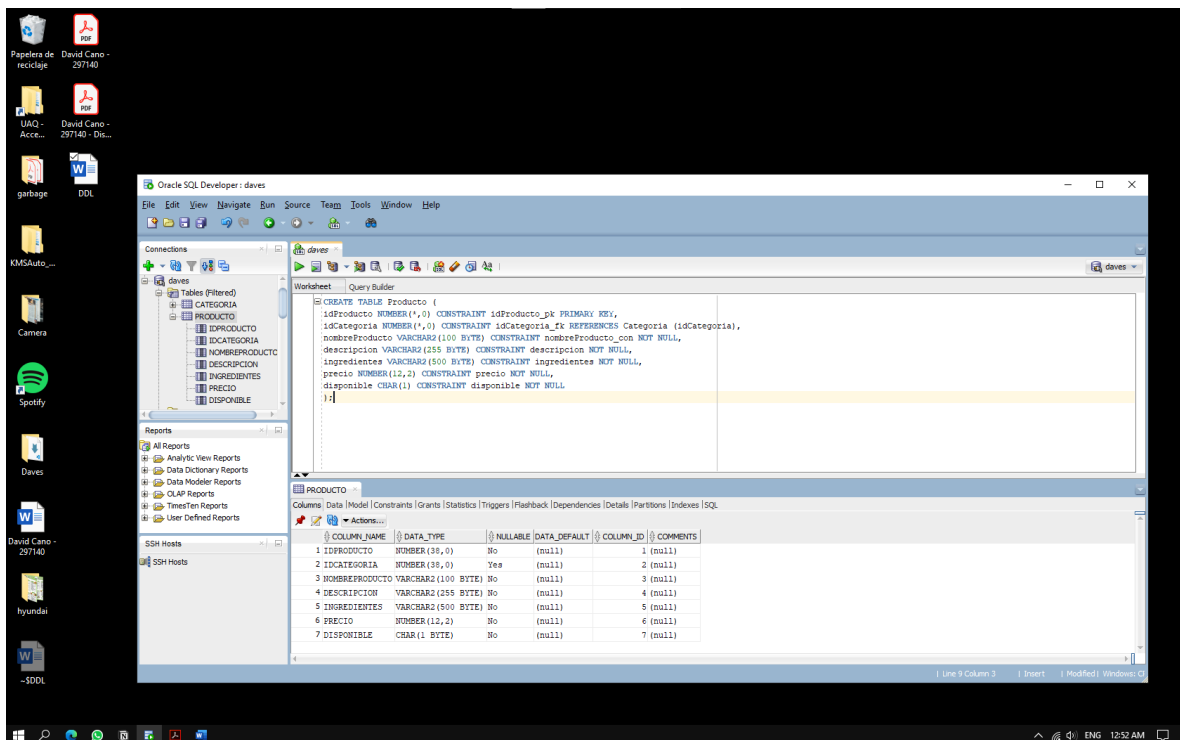
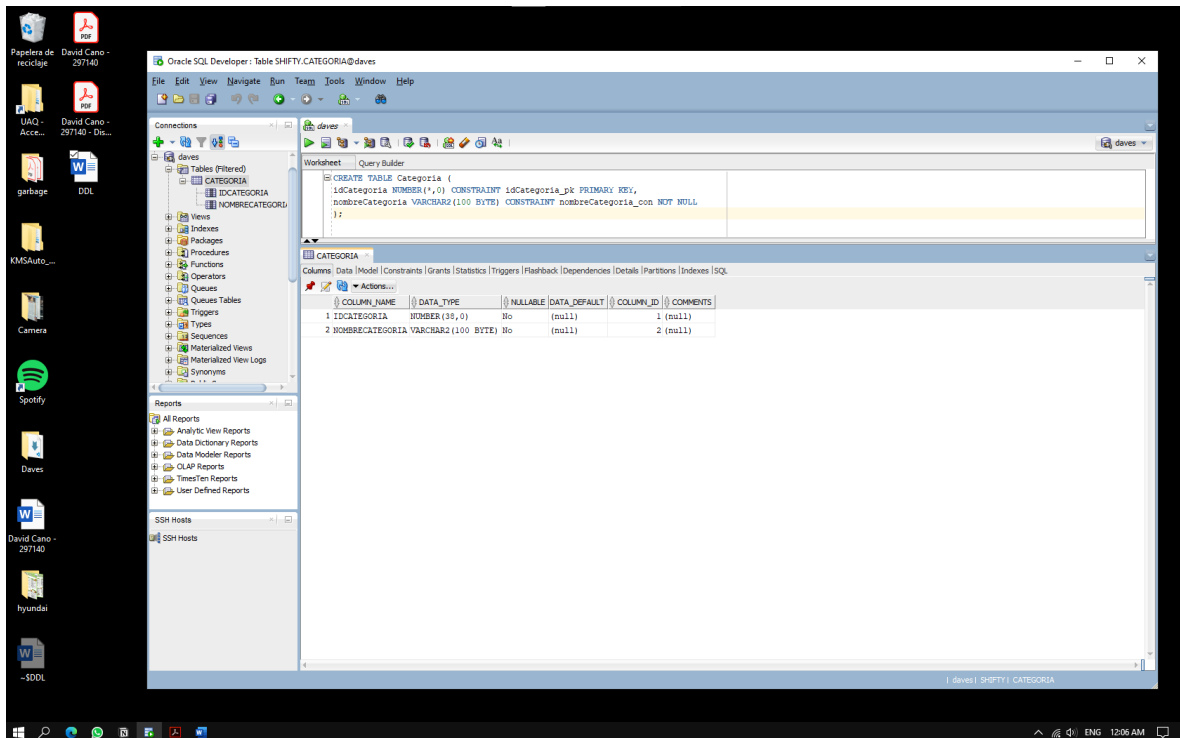
## Diagrama entidad relación con cardinalidad y atributos



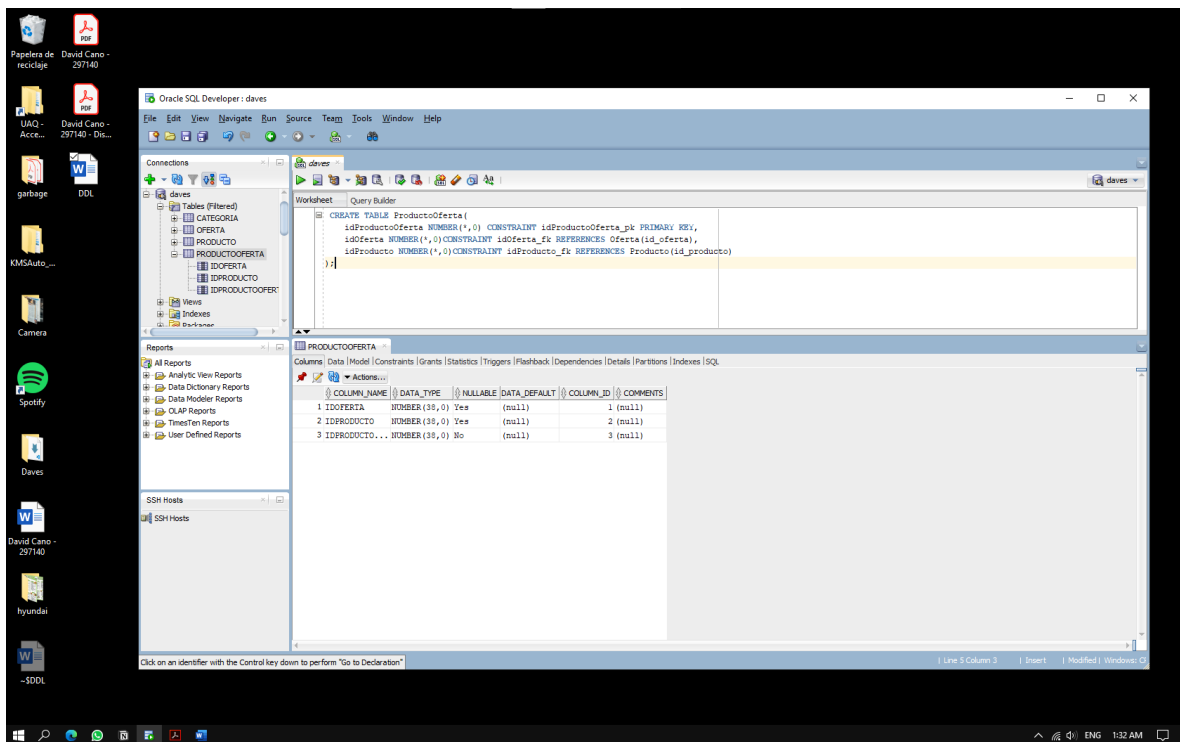
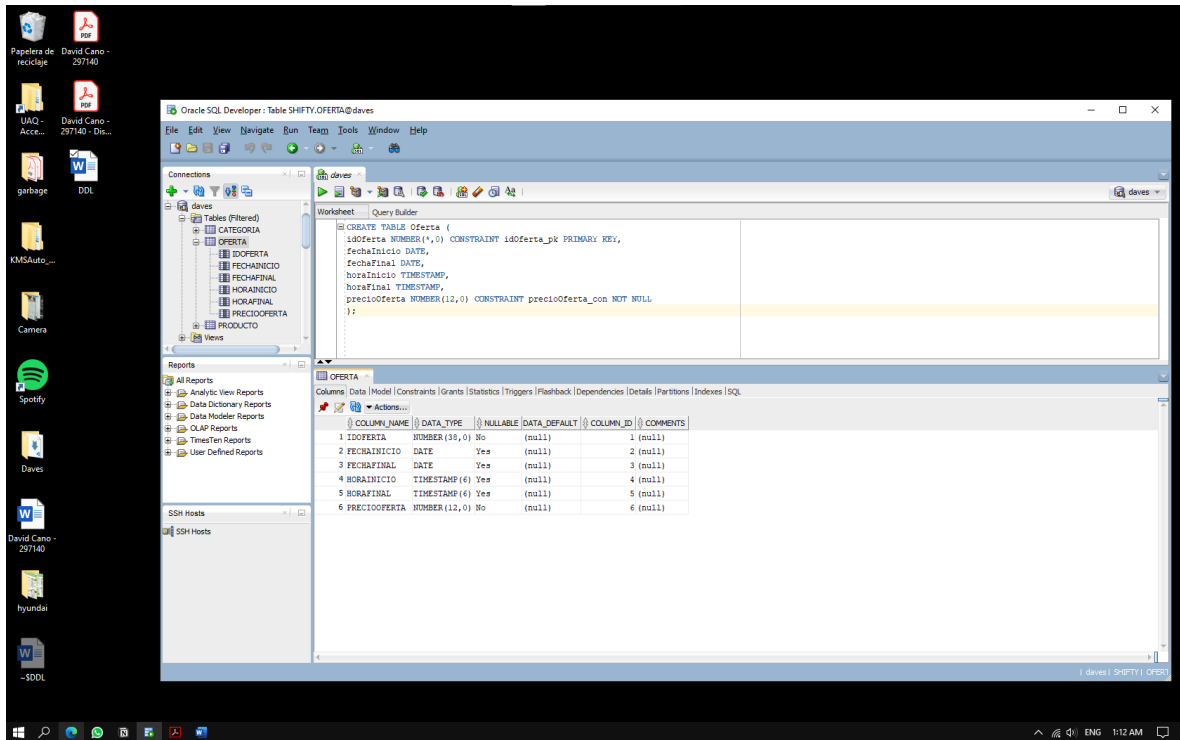
## Modelo físico de datos

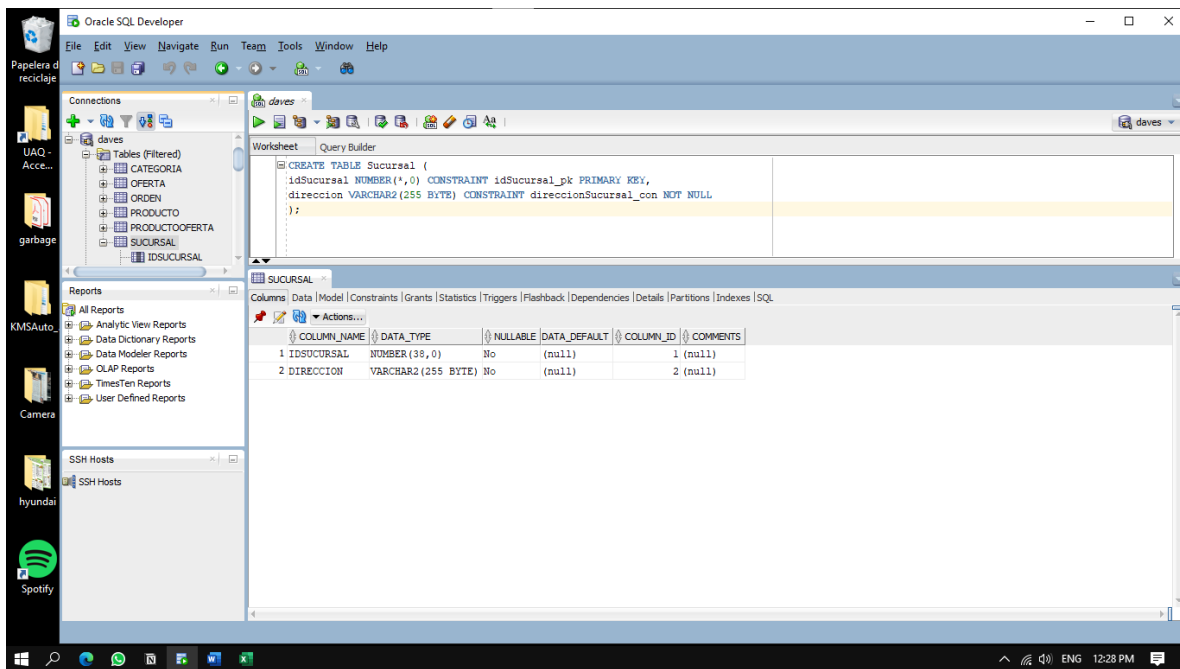
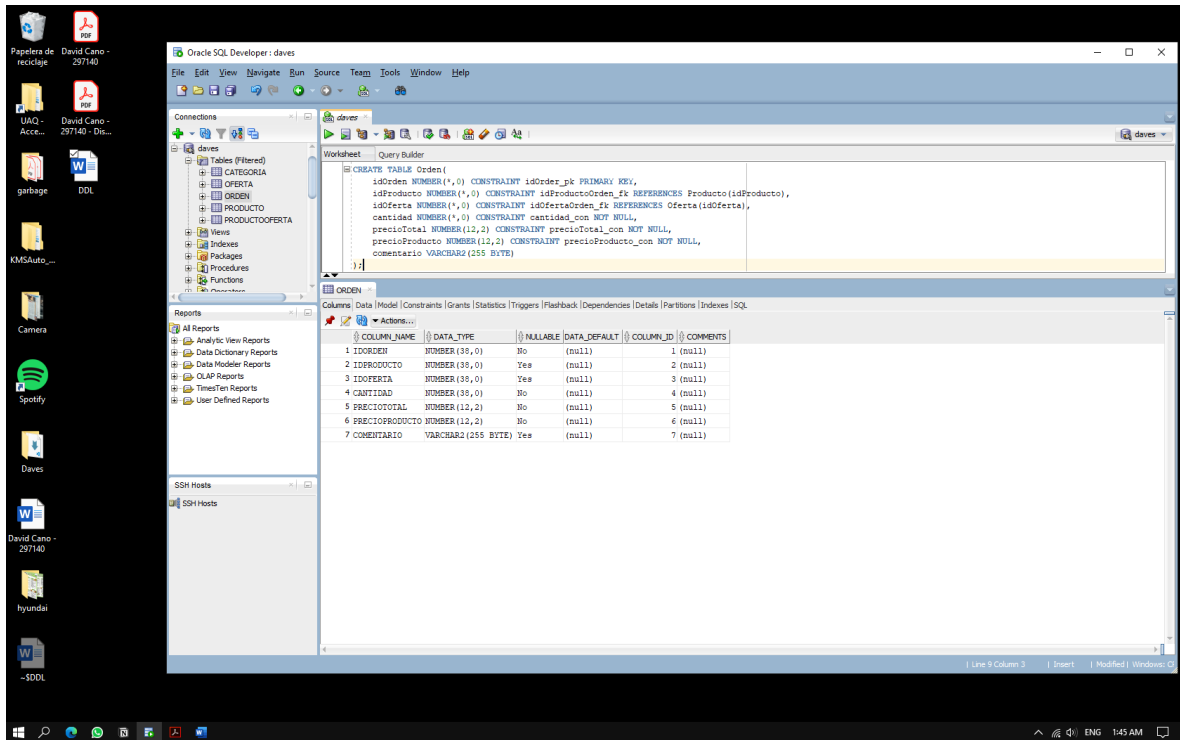


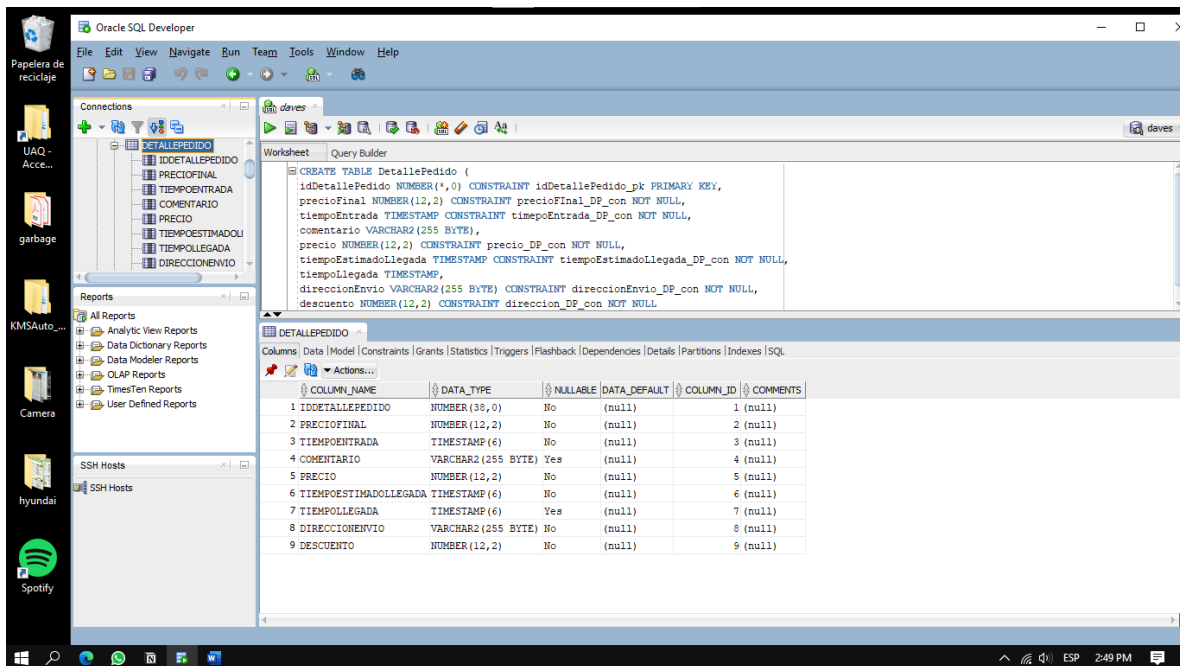
## Sentencias SQL











## Análisis del sistema

### Justificación de la metodología de desarrollo

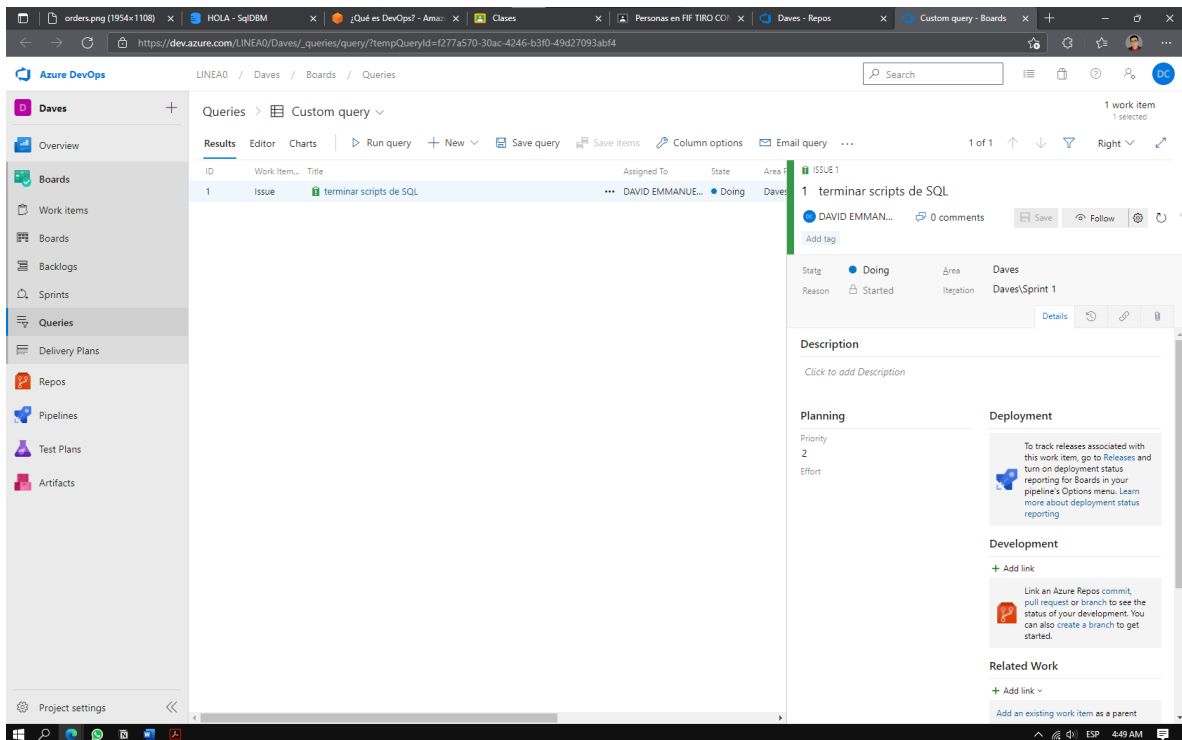
Hoy en día resulta impresionante la cantidad de actividades, procesos y problemas que pueden ser solucionados utilizando un poco de software, sin embargo, ese producto terminado que muchas veces tenemos así de simple frente a nuestros ojos resulta en realidad de una fuerte planificación y cuidado en las características y objetivos que realmente quiere cumplir una solución antes de ser implementada.

Tengo que incluirme dentro de ese grupo de personas que se preocupada siempre exclusivamente por la programación de soluciones (hacer código) y no se preguntaba nunca lo que había detrás. Sin embargo, es gracias a esta materia y al enfoque de Ingeniería de Software fuertemente reforzado durante nuestras lecciones dentro del aula que puedo darme cuenta de lo importante que es una buena planificación y organización al momento de querer sacar adelante un proyecto. Y aún más interesante es lo que se encuentran haciendo las empresas

para adaptarse a las nuevas maneras de hacer código. Sin embargo, antes de hablar de las nuevas hablemos de las viejas: waterfall, prototipado, espiral, incremental... son algunas de las metodologías de desarrollo tradicionales que de acuerdo a Santander (2020), se han convertido en una muy mala idea de implementar si lo que deseamos son resultados flexibles que se adecuen a las necesidades cambiantes del cliente que en primera instancia es quien solicita el software y por el otro lado los desarrolladores que buscan mejorar su productividad a través de lo que ahora se conoce como “metodologías ágiles”.

Ojo, este proyecto no ha implementado una metodología ágil durante sus etapas de planificación análisis y diseño, claro esta que, debido a la documentación, ha sido necesario valernos de las antiguas (pero eficientes) técnicas y procedimientos para resolver el problema objetivo de lo que se esta presentando. Sin embargo, aún existe mucho camino por delante por lo que invitó a trabajar de una manera ágil.

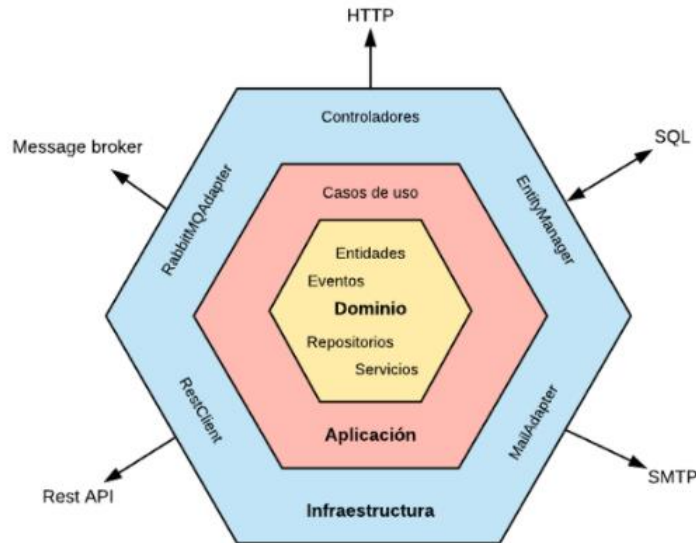
**Individuos e interacciones** sobre procesos y herramientas. **Software funcionando** sobre documentación extensiva. **Colaboración con el cliente** sobre negociación contractual. **Respuesta ante el cambio** sobre seguir un plan. Es en general lo que significa trabajar de una manera ágil, y que como nos indica Red Hat metodologías como **Scrum** o **Kanban** han dado paso a lo que hoy conocemos como **DevOps**



## Diseño del sistema

A continuación, se sugieren algunos patrones de diseño creacionales estructurales y de comportamiento que estandarizan algunos de los problemas más comunes de diseño en el desarrollo y que como indica Sánchez (2020) en su blog funcionan de una excelente manera al momento de querer ir al grano cuando se trata de compartir una solución con un compañero de equipo y que si bien tal vez para esta primera fase del desarrollo no es el caso existirán iteraciones que demanden a un equipo más coordinado.

Antes que nada, me gustaría presentar la **arquitectura hexagonal**. Un patrón centrado en la arquitectura de software que busca de una manera parecida a la arquitectura por capas separar la aplicación en distintas regiones con responsabilidades propias, abstractas y encapsuladas



Y que a grandes rasgos busca definir puertos o puntos de entrada e interfaces, conocidos como **adapters** para que otros módulos, ya sea de UI, Testing etc, puedan implementar y comunicarse con la capa de negocio sin que esta deba saber el origen de la conexión.

## Bibliografía

Briceño, E. A. V. (2013, 27 marzo). *Sistemas de información y su importancia para la empresa*. gestiopolis. <https://www.gestiopolis.com/sistemas-informacion-importancia-empresa/>

Santander Universidades. (2021, 27 octubre). *Metodologías de desarrollo software* / Blog. Becas Santander. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Red Hat. (s. f.). *¿Qué es la metodología ágil?* Recuperado 6 de diciembre de 2021, de <https://www.redhat.com/es/devops/what-is-agile-methodology>

Diví, V. (2020, 30 octubre). *¿Qué es el lenguaje de programación Dart?* inLab FIB. <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>

Sánchez, M. Á. (2020, 30 agosto). *Patrones de Diseño de Software - All you need is Clean Code*. Medium. <https://medium.com/all-you-need-is-clean-code/patrones-de-dise%C3%B1o-b7a99b8525e>