

CODER HOUSE



Torres Rodrigo Emanuel

Gestión de Servicios Web



4 de septiembre de 20XX • Tu ciudad, Provincia

Comision N^a50055

Definición de Modelo de Negocio:

El modelo de negocio Utilizado es una empresa que se dedica a crear y brindar servicios de creación de Software Back-End

Objetivo

Cada día en la agencia se reciben muchos datos para ejercer el servicio de desarrollo y gestión de software Back end. El Objetivo será ordenar los principales datos recibidos en la BbDd, para proporcionar información útil para organizar al equipo y Poder tomar decisiones sobre los servicios ofrecidos. Los datos que principalmente van a ordenarse van a ser “Información de los clientes(Id_cliente,Nombre,Correo,Nº Pedido)”, “Información de trabajos realizados a cada cliente(id_pedido,Nº pedido, nombre-servicio,descripcion_pedido,precio,tiempo_elaboracion, fecha)” y “Cantidad de personas por pedido (id_Servicio,nombre_pedido,area_asignada,personas_asignadas)” . Con esto se podrá mejorar la gestión de los equipos y pulir los procesos necesarios para la oferta del servicio

Objetivos secundarios

Se Busca Conseguir los siguientes Objetivos secundarios:

- Organizar las áreas

Organizar la gestión de Clientes

- Organizar la elaboración de Presupuestos

- Generar un mejor flujo de trabajo

- Gestionar los procesos de cada Servicio Ofrecido

- Administrar de manera efectiva los recursos para cumplir con la mayor cantidad de solicitudes posible

- Tener Registro de los clientes Frecuentes de la Agencia

- Generar una BBDD más compleja que permita recibir de manera ordenada y rápida los datos necesarios

Necesidad que buscamos cubrir:

La necesidad que se busca Cubrir es la falta de datos para la correcta gestión y el correcto ordenamiento general de las distintas áreas encargadas, de los clientes, de los proveedores y de los proyectos

Diagrama entidad-Relación:

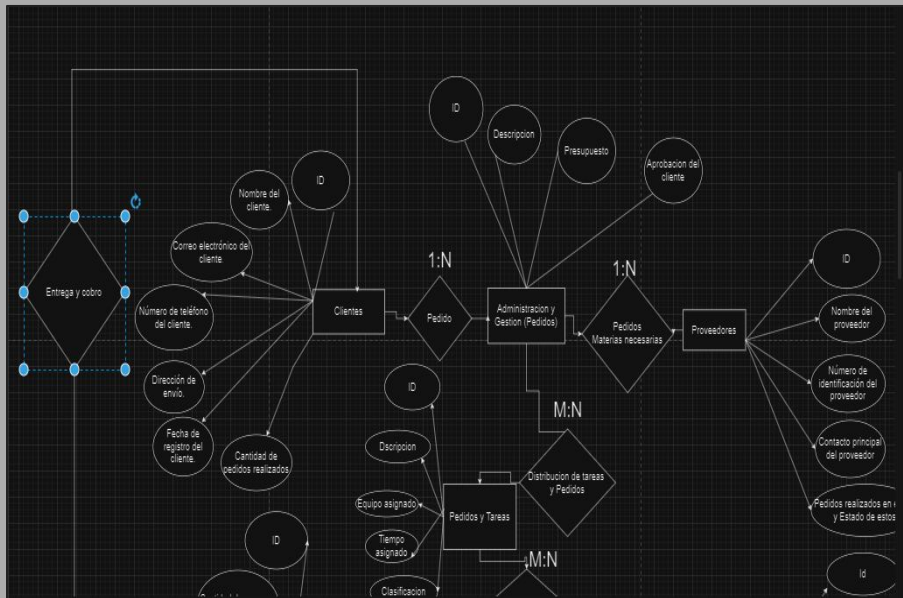
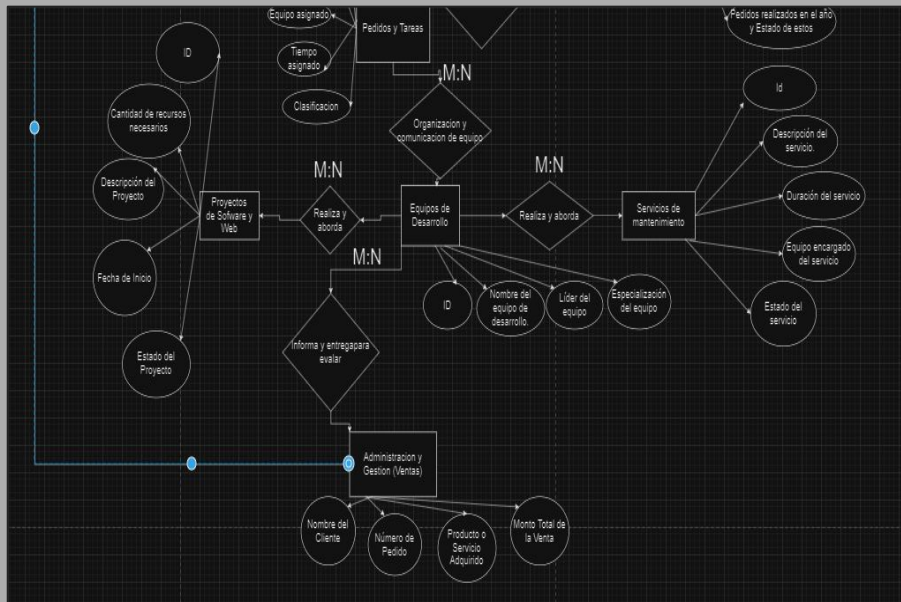


Diagrama Entidad-Relación:



Vistas de la DB

Vista 'vista_clients':

- Objetivo:

- Proporcionar una vista más legible o simplificada de la información de clientes.
- Utiliza alias para cambiar los nombres de las columnas para hacerlos más descriptivos y amigables.

Vista 'vista_horarios':

Objetivo:

- Ofrecer una vista más enfocada en los datos de horarios, extrayendo solo las columnas relevantes.
- Proporciona un conjunto de datos que se centra en la información específica de los horarios de los equipos.

Vista de la DB

Vista 'vista_presu':

- Objetivo:

- Simplificar la presentación de datos de presupuestos, mostrando solo las columnas necesarias.
- Proporciona una vista centrada en los datos clave relacionados con los presupuestos.

Vista 'vista_proyect':

- Objetivo:

- Suministrar una vista más clara de la información de proyectos, extrayendo solo las columnas relevantes.
- Facilita la consulta y presentación de datos específicos sobre proyectos.



Vista 'vista_prov':

- Objetivo:

- Presentar una vista más simple y centrada en los datos de proveedores, utilizando alias para nombres de columnas.
- Proporciona una vista fácilmente comprensible de la información de los proveedores.

Vistas de la DB

vista_presu:

Result Grid   Filter Rows: <input type="text"/>			
	d_pres	v_pres	ok_pres
▶	Presupuesto1	1000	1
	Presupuesto2	1500	0
	Presupuesto3	1200	1
	Presupuesto4	800	0
	Presupuesto5	2000	1
	Presupuesto6	1800	0
	Presupuesto7	900	1
	Presupuesto8	1100	0
	Presupuesto9	1300	1
	Presupuesto10	1600	0

Vistas de la DB

vista_proyect:

	nom_proy	des_proy	f_proy	est_proy	f_fin_proy
►	Proyecto1	Descripción1	2023-01-01	1	2023-03-31
	Proyecto2	Descripción2	2023-02-01	0	2023-04-30
	Proyecto3	Descripción3	2023-03-01	1	2023-05-31
	Proyecto4	Descripción4	2023-04-01	0	2023-06-30
	Proyecto5	Descripción5	2023-05-01	1	2023-07-31
	Proyecto6	Descripción6	2023-06-01	0	2023-08-31
	Proyecto7	Descripción6	2023-07-01	1	2023-09-30
	Proyecto8	Descripción8	2023-08-01	0	2023-10-31
	Proyecto9	Descripción9	2023-09-01	1	2023-11-30
	Proyecto10	Descripción10	2023-10-01	0	2023-12-31

Vistas de la DB

vista_clientes:

Result Grid   Filter Rows: Export:  Wrap Cell Content:

	nom_c	em_c	num_c	Dir_c	id_ped_c
►	Cliente1	cliente1@email.com	1234567890	Nueva Direccion	1
	Cliente2	cliente2@email.com	2345678901	Calle B, Ciudad Y	2
	Cliente3	cliente3@email.com	3456789012	Calle C, Ciudad Z	3
	Cliente4	cliente4@email.com	4567890123	Calle D, Ciudad W	4
	Cliente5	cliente5@email.com	5678901234	Calle E, Ciudad V	5
	Cliente6	cliente6@email.com	6789012345	Calle F, Ciudad U	6
	Cliente7	cliente7@email.com	7890123456	Calle G, Ciudad T	7
	Cliente8	cliente8@email.com	8901234567	Calle H, Ciudad S	8
	Cliente9	cliente9@email.com	9012345678	Calle I, Ciudad R	9
	Cliente10	cliente10@email.com	0123456789	Calle J, Ciudad Q	10

Vistas de la DB

vista_prov:

	nom_prov	cu_prov	num_prov	em_prov
►	Proveedor1	CUIT1	1234567890	proveedor1@email.com
	Proveedor2	CUIT2	2345678901	proveedor2@email.com
	Proveedor3	CUIT3	3456789012	proveedor3@email.com
	Proveedor4	CUIT4	4567890123	proveedor4@email.com
	Proveedor5	CUIT5	5678901234	proveedor5@email.com
	Proveedor6	CUIT6	6789012345	proveedor6@email.com
	Proveedor7	CUIT7	7890123456	proveedor7@email.com
	Proveedor8	CUIT8	8901234567	proveedor8@email.com
	Proveedor9	CUIT9	9012345678	proveedor9@email.com
	Proveedor10	CUIT10	0123456789	proveedor10@email.com

Triggers de la DB

Trigger Before 'before_actualizar_servicio':

- Descripción General: Este trigger se activa antes de realizar una actualización en la tabla 'servicio'.
- Objetivo/Beneficio: Asigna el usuario actual que realiza la actualización, la marca de tiempo actual y actualiza el campo 'estado'.
- Datos/Tablas Manipulados: Se trabaja con la fila que está siendo actualizada en la tabla 'servicio'.

Trigger After 'after_actualizar_servicio':

- Descripción General: Se activa después de actualizar un registro en la tabla 'servicio'.
- Objetivo/Beneficio: Registra en la tabla 'logs' la acción de actualización, el usuario, la fecha y la hora.
- Datos/Tablas Manipulados: Se insertan datos en la tabla 'logs'

Triggers de la DB

Trigger Before 'before_insertar_producto':

- Descripción General: Se activa antes de insertar un nuevo registro en la tabla 'logs'.
- Objetivo/Beneficio: Asigna el usuario actual, la fecha y la hora al insertar un nuevo producto en la tabla 'logs'.
- Datos/Tablas Manipulados: Se trabaja con la fila que está siendo insertada en la tabla 'logs'.

Trigger After 'after_insertar_producto':

- Descripción General: Se activa después de insertar un nuevo registro en la tabla 'logs'.
- Objetivo/Beneficio: Registra en la tabla 'logs' la acción de inserción de un nuevo producto, el usuario, la fecha y la hora.
- Datos/Tablas Manipulados: Se insertan datos en la tabla 'logs'.

Funciones de la DB y su objetivo

- Función: insertar_cliente
 - Descripción: Inserta un nuevo cliente en la tabla 'clients'.
 - Parámetros:
 - - p_nombre: Nombre del cliente (VARCHAR).
 - - p_email: Dirección de correo electrónico del cliente (VARCHAR).
 - - p_numero: Número de teléfono del cliente (VARCHAR).
 - - p_direccion: Dirección del cliente (VARCHAR).
 - - p_id_pedidos: ID de pedidos asociado al cliente (INT).
 - Retorna: El nuevo ID del cliente insertado.

Funciones de la DB y sus objetivos

Función: insertar_proveedor

Descripción: Inserta un nuevo proveedor en la tabla 'proveedores'.

Parámetros:

--p_nombre: Nombre del proveedor (VARCHAR).

--p_cuit: Número de CUIT del proveedor (VARCHAR).

--p_numero: Número de teléfono del proveedor (VARCHAR).

--p_email: Dirección de correo electrónico del proveedor (VARCHAR).

--p_id_pedido_proveedores: ID de pedidos asociado al proveedor (INT).

--Retorna: El nuevo ID del proveedor insertado.

Stored Procedure de la DB

OrdenarTabla

Descripción: Este procedimiento ordena una tabla según un campo y un criterio de ordenamiento dados.

Parámetros:

- tabla_nombre: Nombre de la tabla que se va a ordenar (VARCHAR).
- campo_orden: Campo por el cual se va a ordenar la tabla (VARCHAR).
- ordenamiento: Criterio de ordenamiento (ASC o DESC) (VARCHAR).
- Objetivo/Beneficio: Proporciona una manera flexible de ordenar tablas según diferentes criterios y direcciones.
- Datos/Tablas: Puede manipular cualquier tabla que se pase como argumento.

Stored Procedure de la DB

ActualizarRegistros

Descripción: Este procedimiento actualiza registros en una tabla según una condición dada.

Parámetros:

- tabla_nombre: Nombre de la tabla que se va a actualizar (VARCHAR).
- campo_actualizar: Campo que se va a actualizar en la tabla (VARCHAR).
- nuevo_valor: Nuevo valor que se asignará al campo (VARCHAR).
- condicion: Condición que determina qué registros actualizar (VARCHAR).
- Objetivo/Beneficio: Proporciona una forma dinámica de actualizar registros en una tabla basada en condiciones específicas.
- Datos/Tablas: Puede manipular cualquier tabla que se pase como argumento.