



# Tecnológico de Monterrey

**Campus Queretaro**

**Nombre del Entregable:**

**Evidencia 1 -**

**Análisis de Requerimientos de Software**

**Efectividad en la Negociación**

**Diseño de Componentes de Software**

**Curso:**

**Construcción de Software**

**Profesores:**

**Enrique Alfonso Calderón Balderas**

**Denisse L. Maldonado Flores**

**Alejandro Fernández Vilchis**

**Nombre:**

**Luis Gutierrez Mendioroz | A00837481**

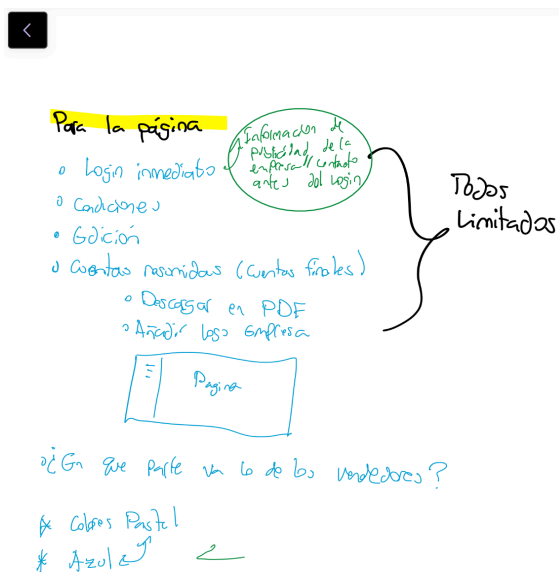
# Desarrollo de Software

¿Cuál fue mi aporte?

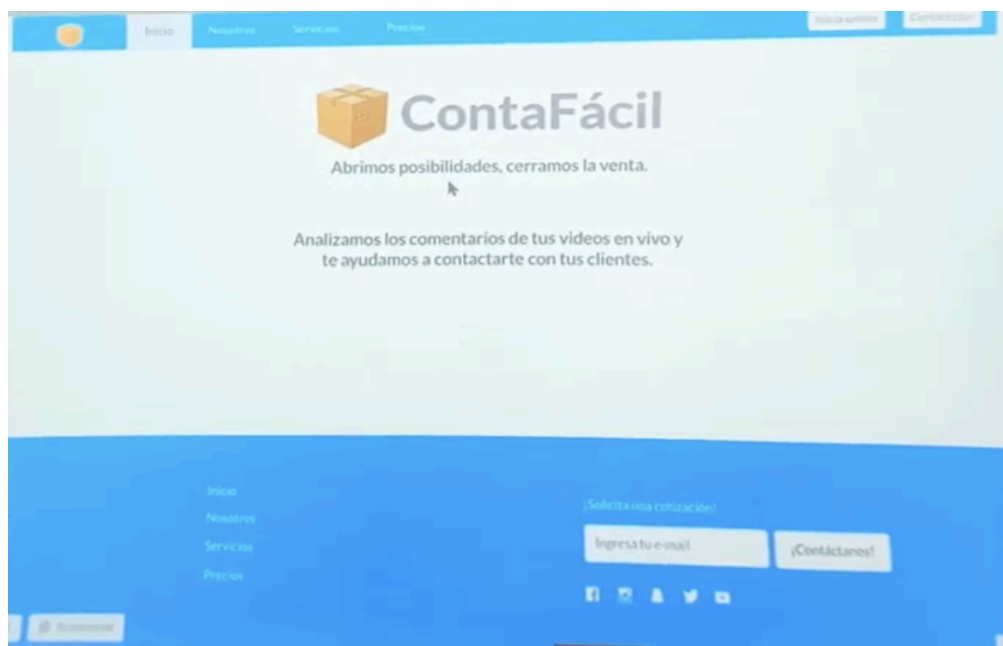
Mi participación consiste en participar activamente en la codificación del proyecto. Mi rol en este proyecto fue el de desarrollador back end por lo que la mayor parte de mis aportes van enfocados a eso. No obstante, estuve participando activamente en parte de la codificación sobre el diseño de la interfaz de nuestra propuesta.

Desarrollo del front-end:

En este apartado no sólo aporté en la codificación sino también el primer prototipo de nuestra interfaz como se muestra en la imagen.



La primera propuesta de interfaz se veía así:



Ahora después de una evolución en diseño se ve así:

Esta evolución se debe a retroalimentación del socio sobre la paleta de colores. Aunque no recibimos los códigos específicos de los colores a utilizar utilice una aplicación de edición gráfica para obtener los códigos en HEX. Obtuve los siguientes:

Hex: F9DECF - Como salmon

Hex: 9ED1EF - Azul telefono

Hex: 848ED0 - Morado telefono y tipografia

Hex: F9E697 - Amarillo rayo



En cuanto a codificación:

Desarrollo del back-end:

Para el desarrollo de nuestra lógica de back-end seguimos el siguiente esquema MVC. Por lo que la codificación está dividida en varias carpetas como: controlador, model, routes, views. Dentro de mis aportaciones destaca dos funciones:

Desarrollo del login el cual no solo permite el acceso sino también identifica el rol del usuario que está intentando acceder

```
if(req.session.role === 1) {
  req.session.usuario = user; // Guardar el usuario en la sesión
  console.log("Usuario es admin --> redirigiendo a admin_view");
  return res.render("usuarios/admin_view.ejs", {
    usuario: user,
    role: req.session.role,
    privileges: req.session.privileges,
  });
}else if(req.session.role === 2) {
  console.log("Usuario es vendedor --> redirigiendo a user_view");
  //return res.redirect('/vendedor'); // Redirigir a la vista del vendedor
  return res.render("usuarios/vendedor_view.ejs", {
    usuario: user,
    role: req.session.role,
    privileges: req.session.privileges,
    csrfToken: req.csrfToken()
  });
};
//else {
```

Despliegue de vendedores que están asignados a una empresa. Para poder implementar dicha funcionalidad tuve que enviar unos comandos de SQL para que el sistema identificará los vendedores asociados a una empresa (la del administrador que haya hecho login) por medio del ID de la compañía.

```
async function findVendedoresByEmpresa(id_Compania) {
  try {
    /*const [rows] = await db.query(
      `SELECT u.ID_User, u.Rol, u.Nombre, u.Nombre_Usuario, u.Password, u.Contacto, e.Nombre_Compania
      FROM usuarios u
      INNER JOIN empresa e ON u.ID_Compania = e.ID_Compania
      WHERE u.Rol = 2 AND e.Nombre_Compania = ?;`,
      [nombre_Compania] //nombre_Compania
    );
    if(!Array.isArray(rows)){
      return rows ? [rows] : [];
    }
    return rows;*/
    const [rows] = await db.query(
      `SELECT u.ID_User, u.Rol, u.Nombre, u.Nombre_Usuario, u.Password, u.Contacto, e.Nombre_Compania
      FROM usuarios u
      INNER JOIN empresa e ON u.ID_Compania = e.ID_Compania
      WHERE u.Rol = 2 AND u.ID_Compania = ?;`,
      [id_Compania]
    );
    return Array.isArray(rows) ? rows : (rows ? [rows] : []);
  }
  catch (error) {
    throw error;
  }
}
```

Esto se despliega con la siguiente vista:

```
<h2>Vendedores de la empresa</h2>
<ul>
  <% if (vendedores && vendedores.length > 0) { %>
    <% vendedores.forEach(function(v) { %>
      <li>Nombre: <%= v.Nombre %> - Nombre Usuario: <%= v.Nombre_Usuario || v.Nombre %> - Contacto: <%= v.Conta
    <% }); %>
  <% } else { %>
    <li>No hay vendedores para esta empresa.</li>
  <% } %>
</ul>
<a href="admin_view.ejs"></a>
```

Link a los commits:

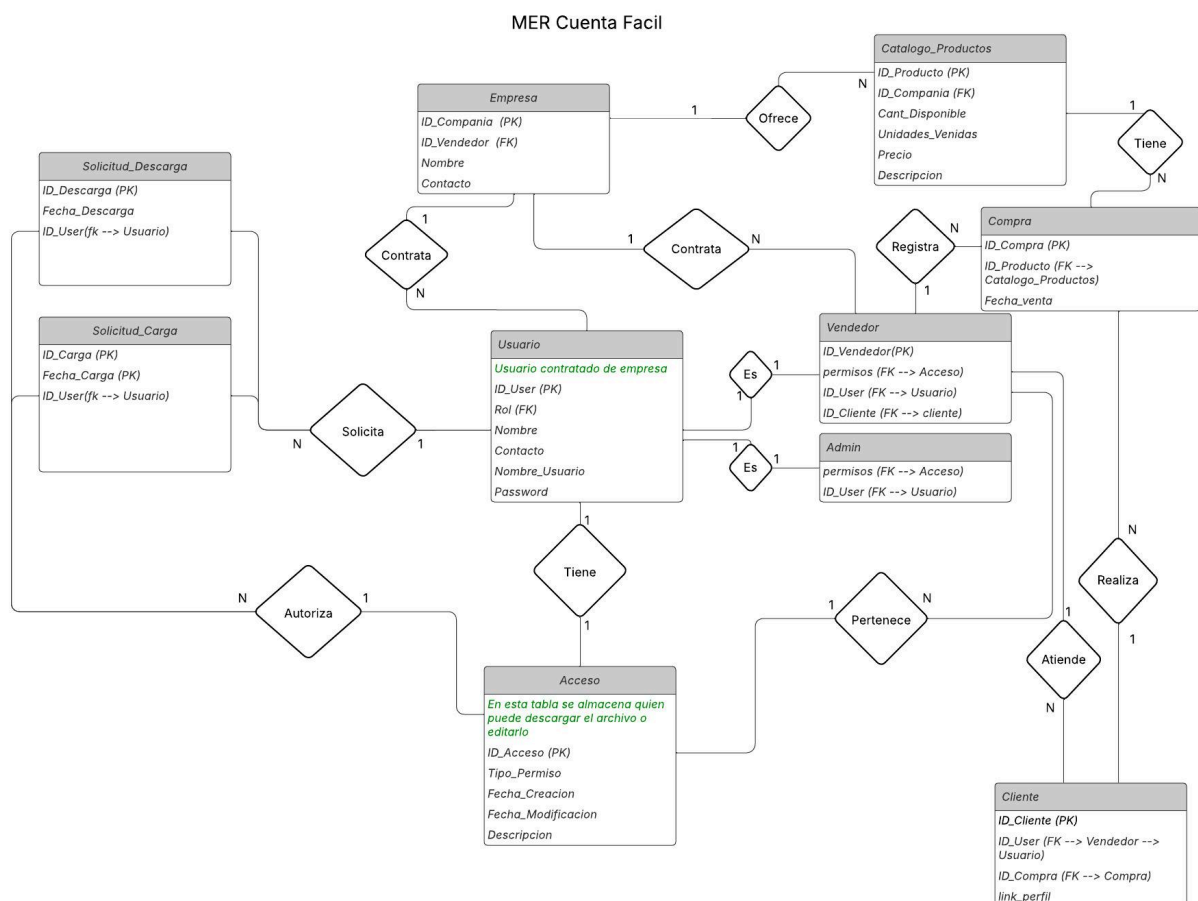
<https://github.com/LuisGutierrez18/ILM-Web-Solutions/commits?author=LuisGutierrez18&before=d9d9a7f2a20600aa6166922f27b45a405e6bf+35>

# Diseño de Componentes de Software

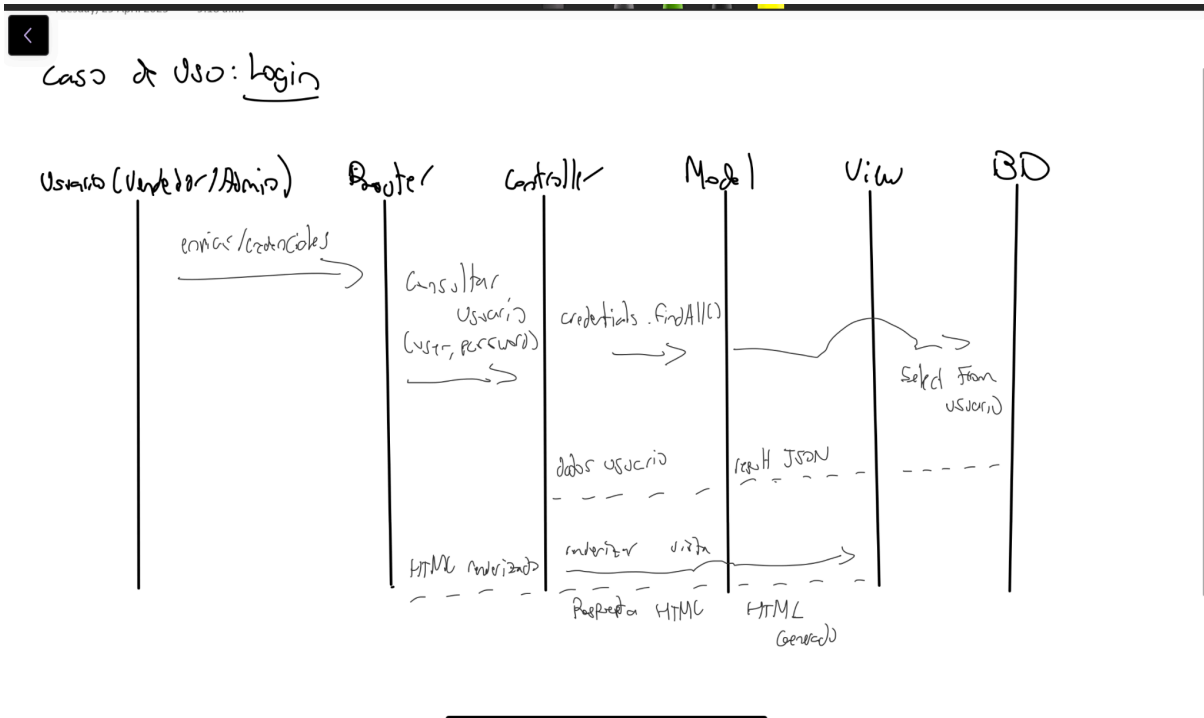
Para esta competencia apoye a mi equipo de la siguiente manera. Contribuí en la creación de diagramas que expliquen el funcionamiento de nuestra página web.

En especial estuve colaborando activamente en la creación del Modelo Entidad Relación. Este modelo, detalla la relación existente entre los diferentes datos dentro de nuestra base de datos. Además explica el funcionamiento que tendrá nuestra base de datos dentro de la página. Ese modelo se elaboró basado en la información recopilada sobre las necesidades de nuestro socio formador.

El mayor reto enfrentado fue transformar nuestras ideas en un modelo lógico debidamente estructurado como lo es el MER. Similarmente, definir de manera correcta las relaciones entre cada entidad, así pudimos asegurar que nuestro diagrama fuera coherente y consistente con la información.



Por otro lado, he estado trabajando en un boceto inicial de un diagrama de secuencia sobre uno de nuestros casos de uso: Login.



Gracias a esto pudimos realizar el MR (modelo relacional) de nuestro sistema:

## Tablas:

### **Empresa**

Id\_Compania (PK)

ID\_Vendedor (FK)

Nombre

Contacto

### **Catalogo\_Productos**

ID\_Producto (PK)

ID\_Compania (FK)

Cant\_Disponible

Unidades\_Vendidas

Precio

Descripcion

### **Compra**

ID\_Compra (PK)

ID\_Compania (FK) → Empresa(ID\_Compania)

Actividad

### **Cliente**

ID\_User (PK)

ID\_User (FK) → Usuario(ID\_User)

ID\_Compra (FK) → Compra(ID\_Compra)

link\_perfil

### **Usuario**

ID\_User (PK)

Rol (FK) → Rol(ID\_Rol)

Nombre

Contacto

Nombre\_Usuario

Password

### **Rol**

ID\_Rol (PK)

Tipo\_Rol

Descripcion

fecha\_alta

Fecha\_Modificacion

### **Vendedor**

ID\_Vendedor (PK)

clientes

ID\_User (FK) → Usuario(ID\_User)

permisos (FK) → Acceso(ID\_Acceso)

### **Admin**

ID\_User (FK) → Usuario(ID\_User)

permisos (FK) → Acceso(ID\_Acceso)

### **Acceso**

ID\_Acceso (PK)

Tipo\_Permiso

fecha\_creacion

fecha\_modificacion



descripcion

### **Solicitud\_Carga**

ID\_Carga (PK)

fecha\_carga

ID\_User (FK) → Usuario(ID\_User)

### **Solicitud\_Descarga**

ID\_Descarga (PK)

fecha\_descarga

ID\_User (FK) → Usuario(ID\_User)