Software Requirements Specifications

LOGIXPRESS

September 2024

0.- Document sheet

Date	Revision	Authors
September - 2024		Mejia Gonzalez David Vazquez Granados Eduardo Antonio Guzman Velazquez Erlin Gabriel

1.INTRODUCTION

This document provides a detailed analysis, including project justification, scope, objectives, requirements, human and technological resources, costs and benefits. It highlights the importance of collections management and efficiency and presents an overview of how this platform can be an effective and customized solution to address these challenges.

1.1 PURPOSE OF THE PROJECT

The purpose of this document is to address one of the most common issues

LOGIXPRESS aims to solve these situations by making deliveries to the places provided by affiliated companies, optimizing resources and time.

LOGIXPRESS offers the treatment of loads, both heavy and specially treated loads, as well as fragile loads, while maintaining efficiency and punctuality.

This Software Requirements Specification (SRS) document follows the IEEE 830 standard and describes in detail the functionalities, features, and limitations of the LOGIXPRESS system. This document will serve as a guide for the development team and provide a clear basis for planning, designing, and implementing the system.

General Description System:

LOGIXPRESS is a logistics management platform designed to optimize the delivery of goods, including heavy, fragile, liquid, and gas loads, while ensuring efficiency and punctuality. The system is built to provide an end-to-end solution for delivery management, from route optimization to vehicle and resource allocation, while maintaining high standards of security and usability.

The platform enables users to manage deliveries by calculating optimal routes, assigning appropriate vehicles based on cargo type and size, and handling resource inventories such as packaging materials. LOGIXPRESS also supports vehicle maintenance tracking, employee management, and the ability to associate specific delivery locations with client accounts. Additionally, it offers features like product tagging and selection of working materials to facilitate proper loading and handling of merchandise.

Key functional areas include user registration, route optimization, vehicle management, delivery tracking, and product handling, all with the aim of improving efficiency, reducing lead times, and providing a customized logistics solution for businesses.

1.2 Scope

This document provides a detailed analysis, including project justification, scope, objectives, requirements, human and technological resources, costs and benefits. It highlights the importance of lead time management and efficiency and is presented as a general overview of how this application can effectively solve and customize to address challenges.

1.3 Personnel involved

NAME	Guzman Velazquez Erlin Gabriel
ROL	Scrum Master
PROFESSIONAL CATEGORY	Student
RESPONSIBILITIES	Data base
CONTACT INFORMATION	0323105859@ut-tijuana.edu.mx
APPROVAL	

NAME	Mejia Gonzales David
ROL	Development Team
PROFESSIONAL CATEGORY	Student
RESPONSIBILITIES	Data base
CONTACT INFORMATION	0322103760@ut-tijuana.edu.mx
APPROVAL	

NAME	Vazquez Granados Eduardo Antonio
ROL	Product Owner
PROFESSIONAL CATEGORY	Student
RESPONSIBILITIES	Data base
CONTACT INFORMATION	0322103840@ut-tijuana.edu.mx
APPROVAL	

1.4 Definitions, acronyms and abbreviations

Nombre	Description
User	Person who will use the system to manage the data
RF	Functional Requirement
SRS	Software Requirements Specification
NFR	Non-functional requirement

1.5 References

Document Title	Reference
Standard IEEE 830- 1998	IEEE

1.6 Overview

In the program we expect to optimize time and resources for the transportation of shipments by customers. We will implement an architecture that will help us to avoid system collapse, we will also monitor the process to ensure that it has a high reliability, and we will also put a priority focus on security. All this in order to achieve a reliable delivery agreement between us and the customer.

This document is fundamental for understanding LOGIXPRESS and will be useful to everyone involved in its development. It will clearly and structurally explain the requirements and objectives for achieving an efficient and effective development. 1.7 Funtional Requerements

2.- Functional Requirements

1. Functional Requirement: User Registration

Description:
The system will allow new users to register by entering an email address, full name, phone number, and password.

Inputs:
- Full Name
- Company
- Email Address
- Phone Number
- Password

Outputs:
- Successful registration confirmation
- Error message if registration fails

2. Functional Requirement: User Login

Description:
The system will allow registered users to log in.

Inputs:
- Email Address
- Password

Outputs:
- Access if credentials are valid
- Error message if credentials are invalid

3. Functional Requirement: User Account Management

Description:
The user will be able to manage their account information, including updating data, locations, and deleting their account.

Inputs:
- Full Name
- Company
- Email Address
- Phone Number
- Password
- Locations

Outputs:
- Updated data

4. Functional Requirement: Route Optimization

Description:
The system will calculate optimal routes for a set of deliveries.

Inputs:
- Delivery Destination
- Delivery Origin
- Time Window

Outputs:
- Optimized route
- Estimated distance and time

5. Functional Requirement: Registration and Removal of Freight Vehicles

Description:
The system will allow for the registration and removal of freight vehicles.

Inputs:
- For Registration:
  - Serial Number
  - Select Model
  - Select Brand
  - Select Load Type
  - Select Vehicle Category
  - Mileage
  - Fuel consumption per kilometer
- For Removal:
  - Vehicle Number

Outputs:
- For Registration:
  - Successful registration message
- For Removal:
  - Successful removal message

6. Functional Requirement: Cargo Vehicle Assignment

Description:
Available vehicles will be managed for deliveries, selecting the most suitable one based on the type, weight, and size of the cargo.

Inputs:
- Vehicle Number

Outputs:
- Vehicle Number
- Vehicle Type
- License Plate
- Serial Number
- Availability
- Status

7. Functional Requirement: Delivery Management

Description:
Control and tracking of deliveries will be managed.

Inputs:
- Delivery Number
- Status
- Scheduling Window

Outputs:
- Delivery Number
- Delivery status update

8. Functional Requirement: Employee Management

Description:
Employee assignment to deliveries will be managed.

Inputs:
- Employee Number

Outputs:
- Employee Number
- First Name
- Last Name

9. Functional Requirement: Location Management

Description:
The client can associate locations with their account.

Inputs:
- Code
- Address

Outputs:
- Address

10. Functional Requirement: Product

Description:
The client will specify the approximate weight, dimensions, quantity, labeling (Fragile or General Load), and product category (Cold Cuts, Appliances, Construction Materials, etc.).

Inputs:
- Code
- Weight
- Dimensions
- Quantity
- Brand
- Status

Outputs:
- Weight
- Dimensions
- Quantity
- Brand

11. Functional Requirement: Vehicle Maintenance Management

Description:
Maintenance costs, dates, and descriptions of work performed on vehicles will be specified.

Inputs:
- Code
- Description
- Cost
- Date

Outputs:
- Description
- Cost
- Date

12. Functional Requirement: Vehicle Usage History

Description:
Displays the usage information for a specific vehicle.

Inputs:
- Vehicle Number

Outputs:
- Vehicle Number
- Employee Name
- Usage Date
- Delivery
- Mileage before trip
- Mileage after trip

3.- No Functional Requirements

RNF1: Data Security

- Description: The system must ensure the security and confidentiality of user data through end-to-end encryption and robust security measures.

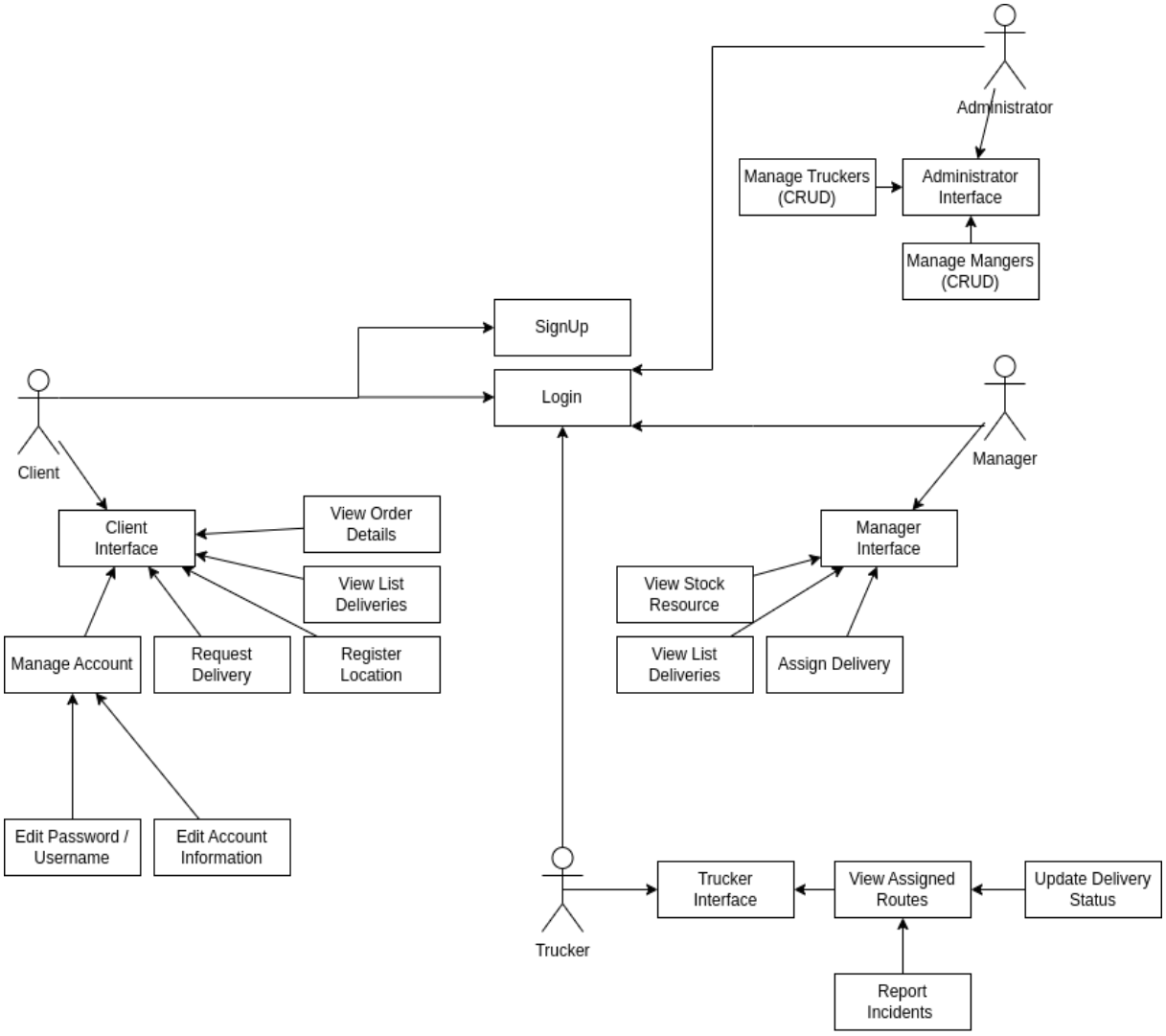
RNF2: Availability

- Description: The system must be available 24 hours a day, 7 days a week, with minimal planned downtime for maintenance.

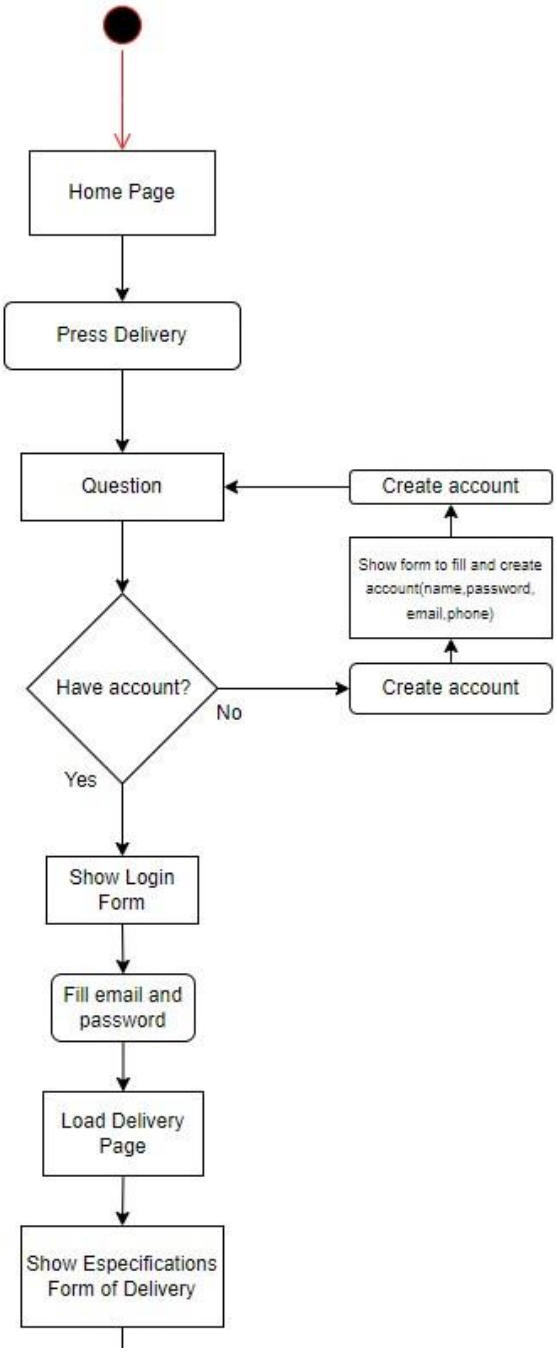
RNF3: Usability

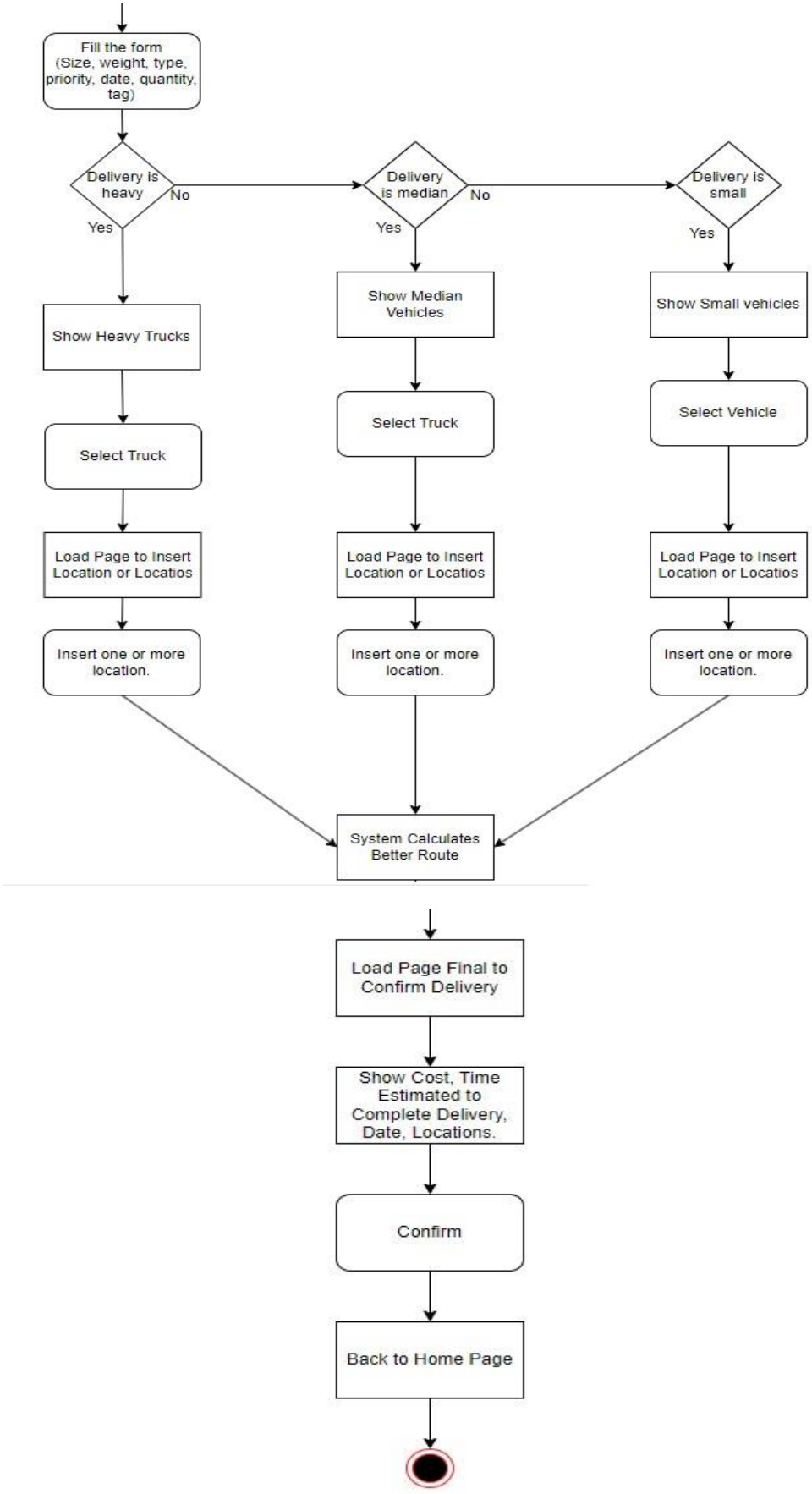
- Description: The system must have an intuitive and user-friendly interface, both on the website and the mobile application, to ensure user convenience

Use of Cases

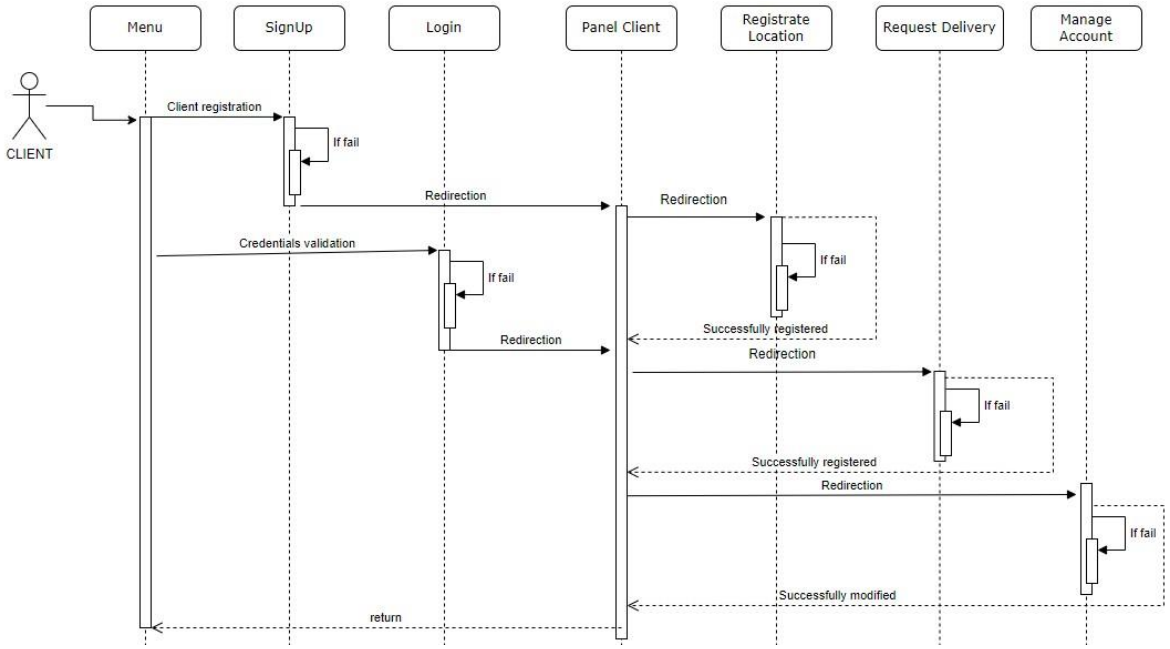

Administrator
Manage Truckers (CRUD)
Administrator Interface
Manage Mangers (CRUD)
SignUp
Login
Client
Manager
Client Interface
View Order Details
View List Deliveries
Register Location
Request Delivery
Manage Account
Edit Password / Username
Edit Account Information
Manager Interface
View Stock Resource
View List Deliveries
Assign Delivery
Trucker
Trucker Interface
View Assigned Routes
Update Delivery Status
Report Incidents

Activities

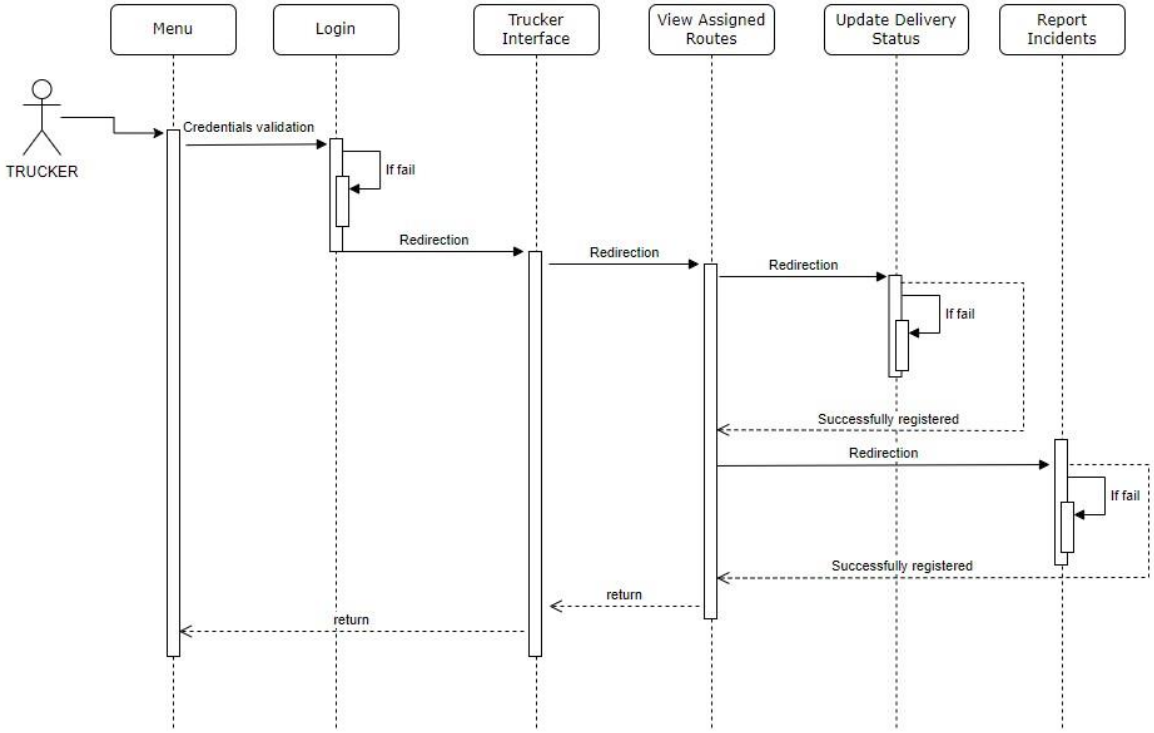

Home Page
Press Delivery
Question
Create account
Show form to fill and create account(name,password, email,phone)
Have account?
No
Create account
Yes
Show Login Form
Fill email and password
Load Delivery Page
Show Especifications Form of Delivery


Fill the form (Size, weight, type, priority, date, quantity, tag)
Delivery is heavy
No
Yes
Delivery is median
No
Yes
Delivery is small
Yes
Show Heavy Trucks
Select Truck
Load Page to Insert Location or Locatios
Insert one or more location.
Show Median Vehicles
Select Truck
Load Page to Insert Location or Locatios
Insert one or more location.
Show Small vehicles
Select Vehicle
Load Page to Insert Location or Locatios
Insert one or more location.
System Calculates Better Route
Load Page Final to Confirm Delivery
Show Cost, Time Estimated to Complete Delivery, Date, Locations.
Confirm
Back to Home Page

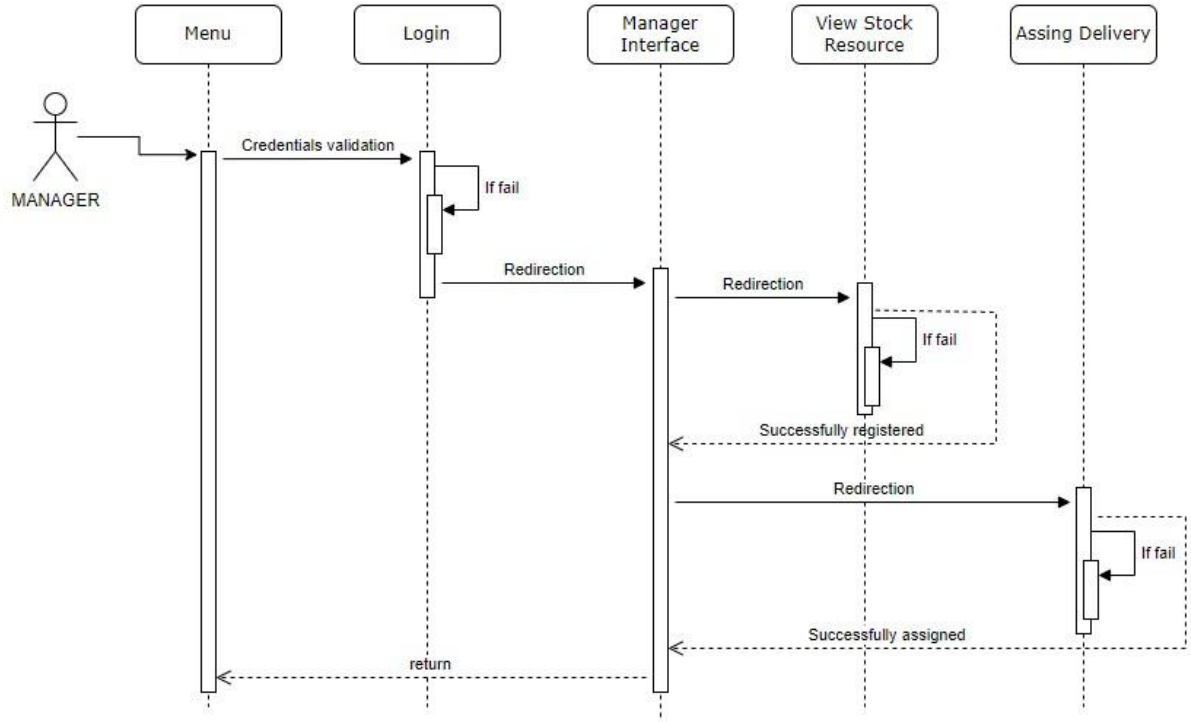
Sequence Client



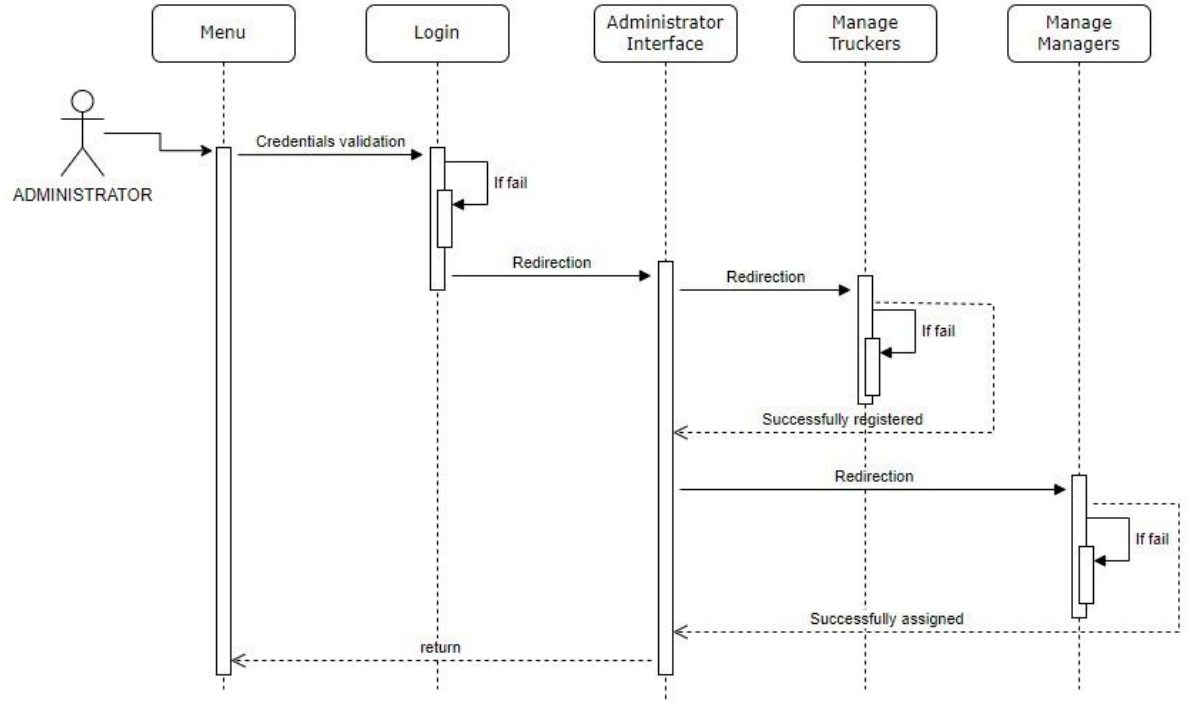
Sequency Trucker



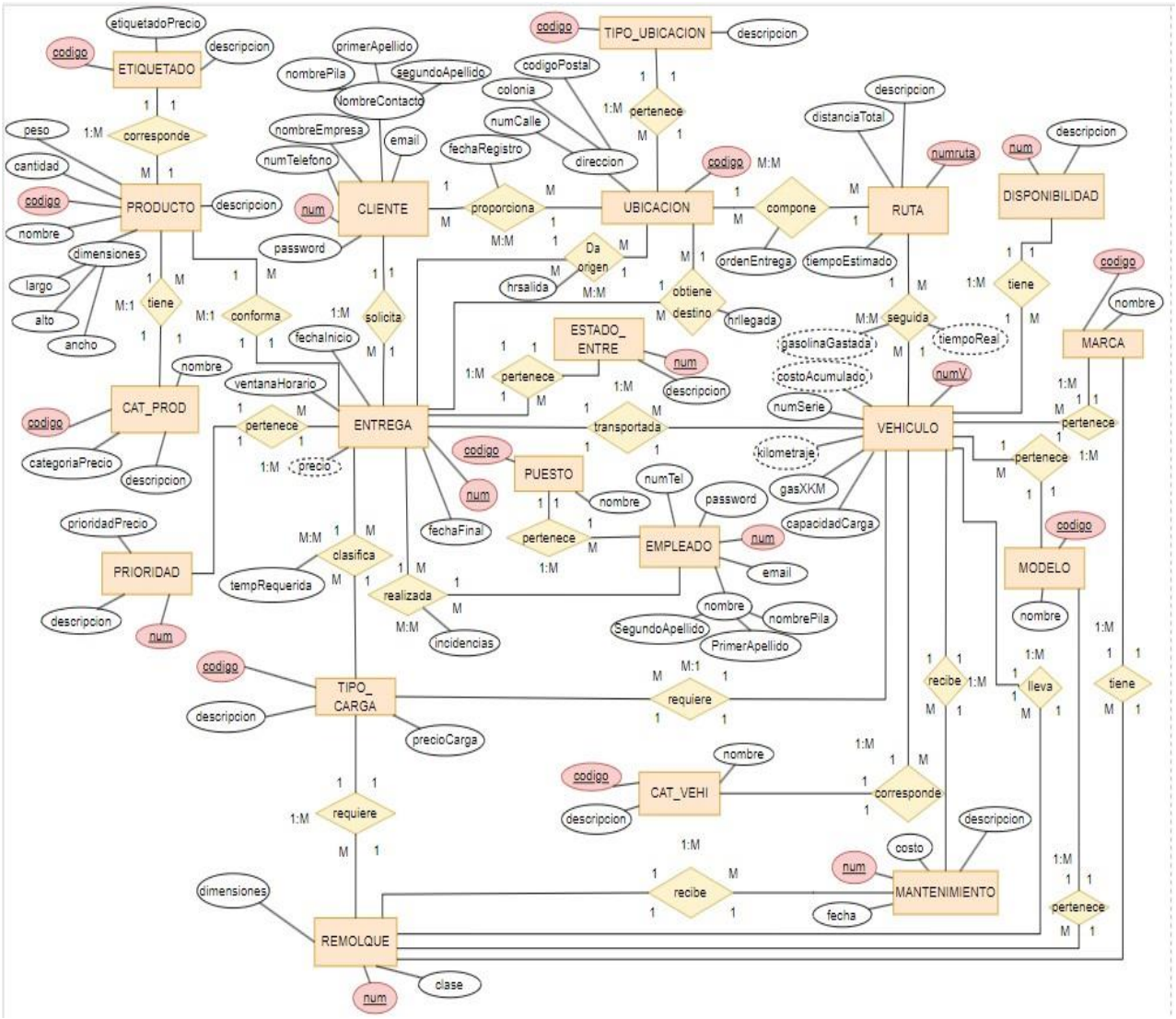
Sequency Manager



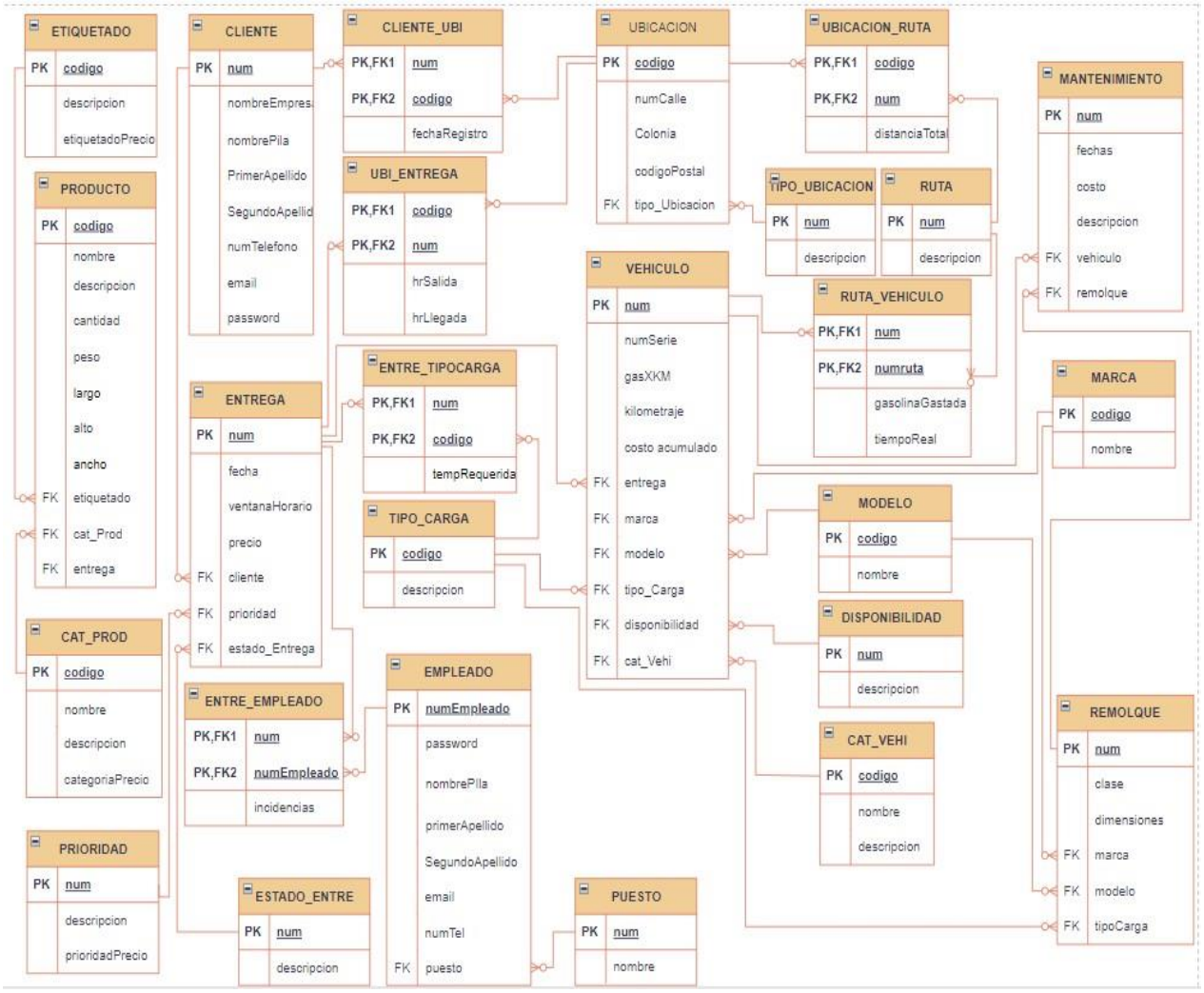
Sequency Administrator



DER


etiquetadoPrecio
codigo
ETIQUETADO
descripcion
corresponde
peso
cantidad
codigo
PRODUCTO
descripcion
nombre
dimensiones
largo
alto
ancho
tiene
conforma
nombre
CAT_PROD
codigo
categoriaPrecio
descripcion
prioridadPrecio
PRIORIDAD
descripcion
num
primerApellido
nombrePila
segundoApellido
NombreContacto
nombreEmpresa
email
numTelefono
num
CLIENTE
password
solicita
fechaInicio
ventanaHorario
pertenece
precio
ENTREGA
fechaFinal
clasifica
tempRequerida
realizada
incidencias
codigo
TIPO_UBICACION
descripcion
codigoPostal
colonia
numCalle
direccion
pertenece
fechaRegistro
proporciona
UBICACION
Da origen
hrsalida
obtiene destino
hrllegada
codigo
compone
ordenEntrega
distanciaTotal
descripcion
numruta
RUTA
tiempoEstimado
seguida
gasolinaGastada
tiempoReal
costoAcumulado
num
DISPONIBILIDAD
descripcion
tiene
codigo
MARCA
nombre
ESTADO_ENTRE
num
descripcion
pertenece
transportada
codigo
PUESTO
nombre
num
pertenece
numTel
password
EMPLEADO
num
email
nombre
nombrePila
SegundoApellido
PrimerApellido
numV
numSerie
kilometraje
gasXKM
capacidadCarga
VEHICULO
pertenece
codigo
MODELO
nombre
tiene
lleva
recibe
codigo
TIPO_CARGA
descripcion
precioCarga
requiere
codigo
CAT_VEHI
descripcion
nombre
corresponde
requiere
dimensiones
REMOLQUE
num
clase
recibe
num
fecha
costo
descripcion
MANTENIMIENTO
pertenece

MR


ETIQUETADO
PK codigo
descripcion
etiquetadoPrecio
CLIENTE
PK num
nombreEmpresa
nombrePila
PrimerApellido
SegundoApellid
numTelefono
email
password
CLIENTE_UBI
PK,FK1 num
PK,FK2 codigo
fechaRegistro
UBICACION
PK codigo
numCalle
Colonia
codigoPostal
FK tipo_Ubicacion
UBICACION_RUTA
PK,FK1 codigo
PK,FK2 num
distanciaTotal
MANTENIMIENTO
PK num
fechas
costo
descripcion
FK vehiculo
FK remolque
PRODUCTO
PK codigo
nombre
descripcion
cantidad
peso
largo
alto
ancho
FK etiquetado
FK cat_Prod
FK entrega
UBI_ENTREGA
PK,FK1 codigo
PK,FK2 num
hrSalida
hrLlegada
TIPO_UBICACION
PK num
descripcion
RUTA
PK num
descripcion
VEHICULO
PK num
numSerie
gasXKM
kilometraje
costo acumulado
FK entrega
FK marca
FK modelo
FK tipo_Carga
FK disponibilidad
FK cat_Vehi
RUTA_VEHICULO
PK,FK1 num
PK,FK2 numruta
gasolinaGastada
tiempoReal
MARCA
PK codigo
nombre
ENTREGA
PK num
fecha
ventanaHorario
precio
FK cliente
FK prioridad
FK estado_Entrega
ENTRE_TIPOCARGA
PK,FK1 num
PK,FK2 codigo
tempRequerida
TIPO_CARGA
PK codigo
descripcion
MODELO
PK codigo
nombre
DISPONIBILIDAD
PK num
descripcion
CAT_PROD
PK codigo
nombre
descripcion
categoriaPrecio
EMPLEADO
PK numEmpleado
password
nombrePila
primerApellido
SegundoApellido
email
numTel
FK puesto
ENTRE_EMPLEADO
PK,FK1 num
PK,FK2 numEmpleado
incidencias
CAT_VEHI
PK codigo
nombre
descripcion
REMOLQUE
PK num
clase
dimensiones
FK marca
FK modelo
FK tipoCarga
PRIORIDAD
PK num
descripcion
prioridadPrecio
ESTADO_ENTRE
PK num
descripcion
PUESTO
PK num
nombre

4.- STAGES OF WEB PAGE CREATION

1. PRE-PROCESS

In this stage, the roles for the process of building the website are assigned, assigning who will be in charge of each process.

Mejia Gonzalez David	Creador de contenido
Vázquez Granados Eduardo Antonio	Arquitecto de la información
Mejia Gonzalez David	Pruebas
Vázquez Granados Eduardo Antonio	Diseñador grafico
Guzmán Velázquez Erlin Gabriel	Diseñador UI y UX
Guzmán Velázquez Erlin Gabriel	Front-End
Vázquez Granados Eduardo Antonio	Back-End
Mejia Gonzalez David	Supervisor de Testing

2. CONTENT STRATEGY

Website planning, development and administration

- This phase addresses content planning for order management and delivery optimization. The goal is to provide a clear and functional experience for users interacting with the system.
- Overview: The main purpose of the website is to help customers manage, track, and optimize their deliveries. Identify audiences (businesses, individuals) and their specific needs, such as ease of ordering, real-time tracking, and route optimization.
- **Content plan:** Plan the types of content that will be needed, such as:
  - **Description of services:** Detailed explanation of delivery services, order management and route optimization process.
  - **Guides and FAQs:** Create content that makes the system easier to use, such as tutorials for placing orders, tracking deliveries, and managing logistics.
  - **Testimonials or case studies:** Show success stories of companies or individuals who have used the service effectively. Content should focus on the clarity, reliability, and efficiency of the system.

Website Content

- Communicate the logistics process and provide the necessary tools for customers to manage their deliveries effectively. Clear instructions, defined action buttons, and real-time updates should be the pillars of the content.
- **Visitor view:** Identify that the most important thing for site users will be efficient order management, transparency in delivery times and route optimization.
- **Who you want to reach and who they are:** The following is identified as the target audience:
  - **Companies:** Those that require the transport of goods, such as construction materials, furniture or food.
- **Audit images, videos, texts, and how content relates to another:**
  - **Images:** Images should show examples of products transported, vehicles used, and tracking tools.
  - **Texts:** Texts should be simple and clear, explaining each step of the delivery process. The relationship between the contents should be logical, guiding the user through the entire order management process.

Content Modeling

- **Hierarchy**
  - **Most important pages:** The main pages include the order form, order tracking, and delivery options.
  - **Support pages:** Pages that provide additional information, such as FAQs and testimonials, should complement the main pages.
  - **Supplemental pages:** Pages such as contact, and company information should provide support for key pages.
- **Information Architecture:** Structure content logically, ensuring clear navigation between order management, tracking, and service details sections.
- **Wireframing:** Design wireframes to visualize the arrangement of elements on the page, without focusing on colors yet. Place forms, buttons, and content blocks in their general positions, and add explanatory comments.

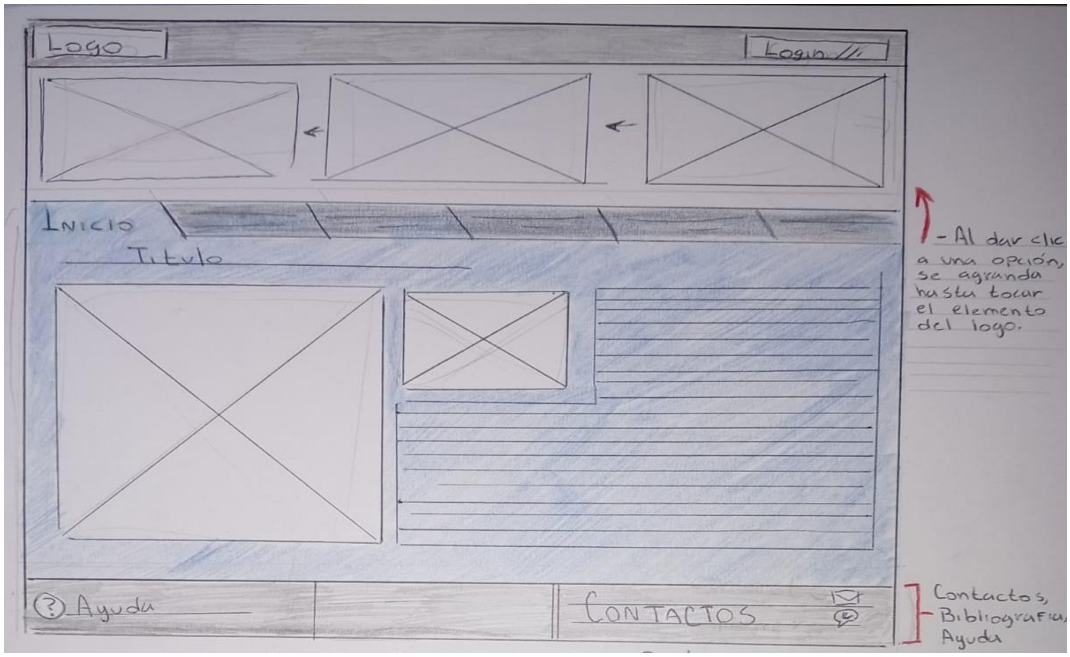


Figure 1 Home Concept

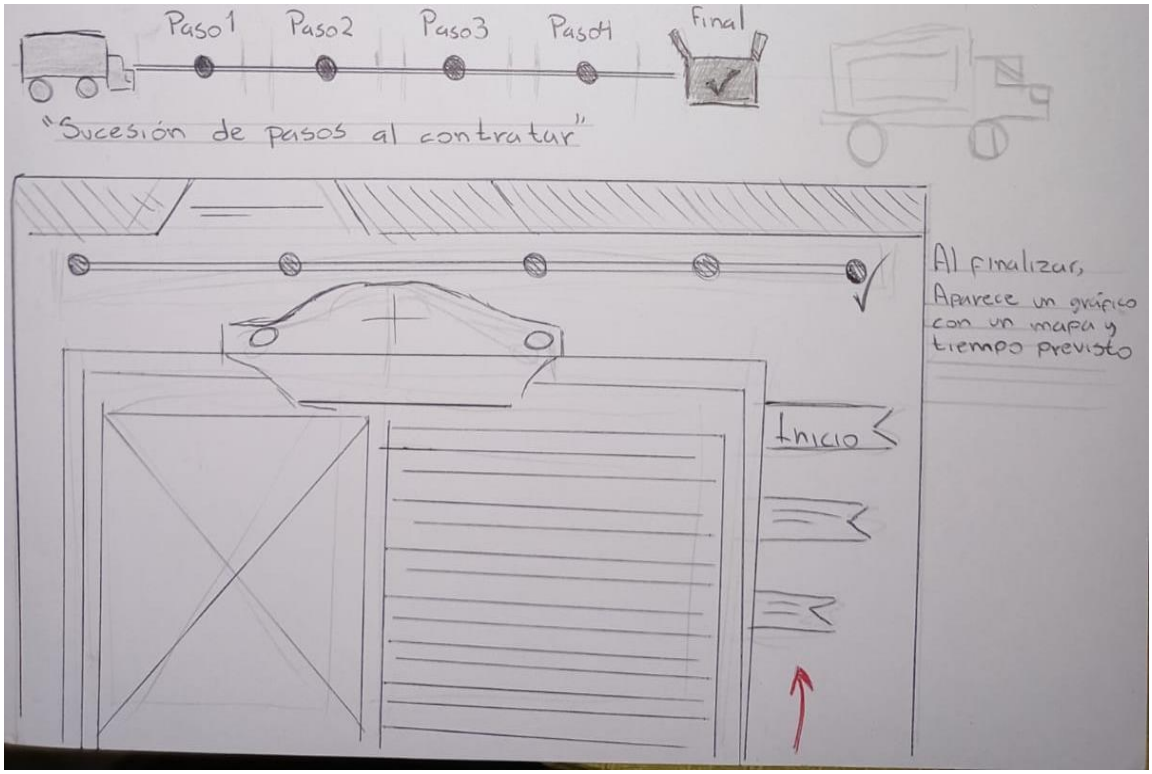


Figure 2 Delivery Request Form Page Concept

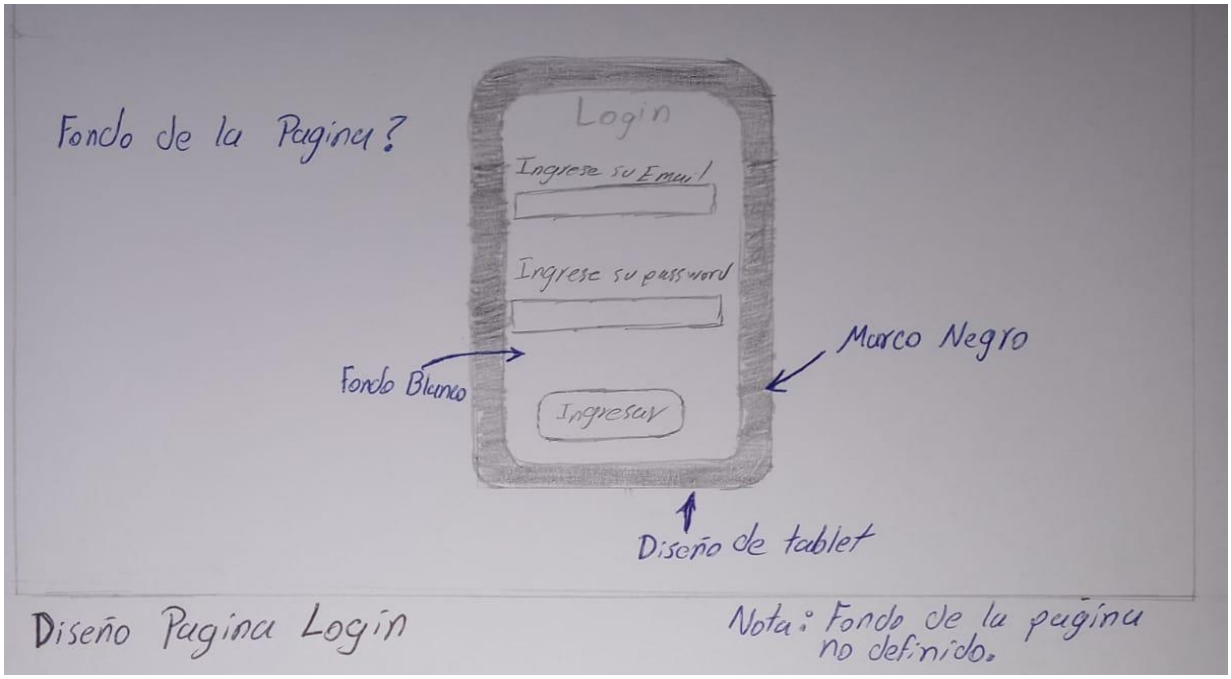


Figure 3 Login Page Concept

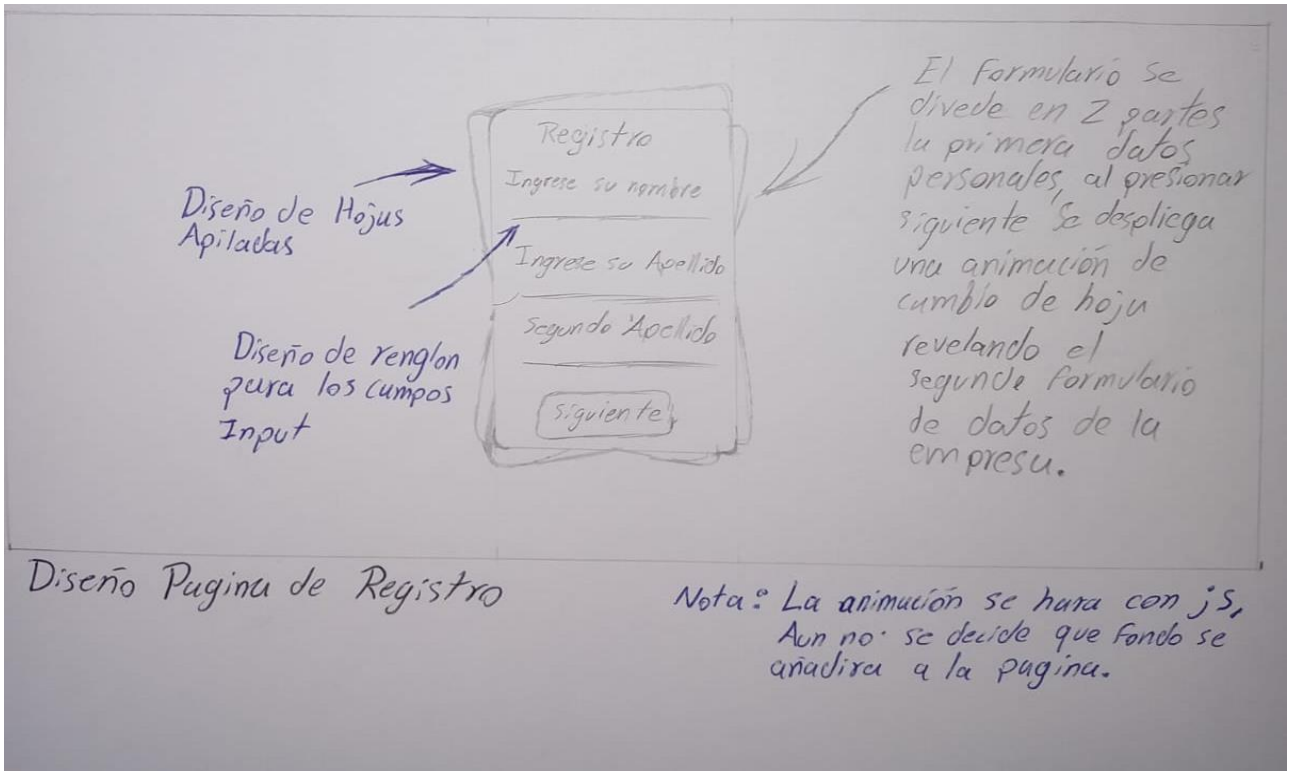


Figure 4 Register Page Concept

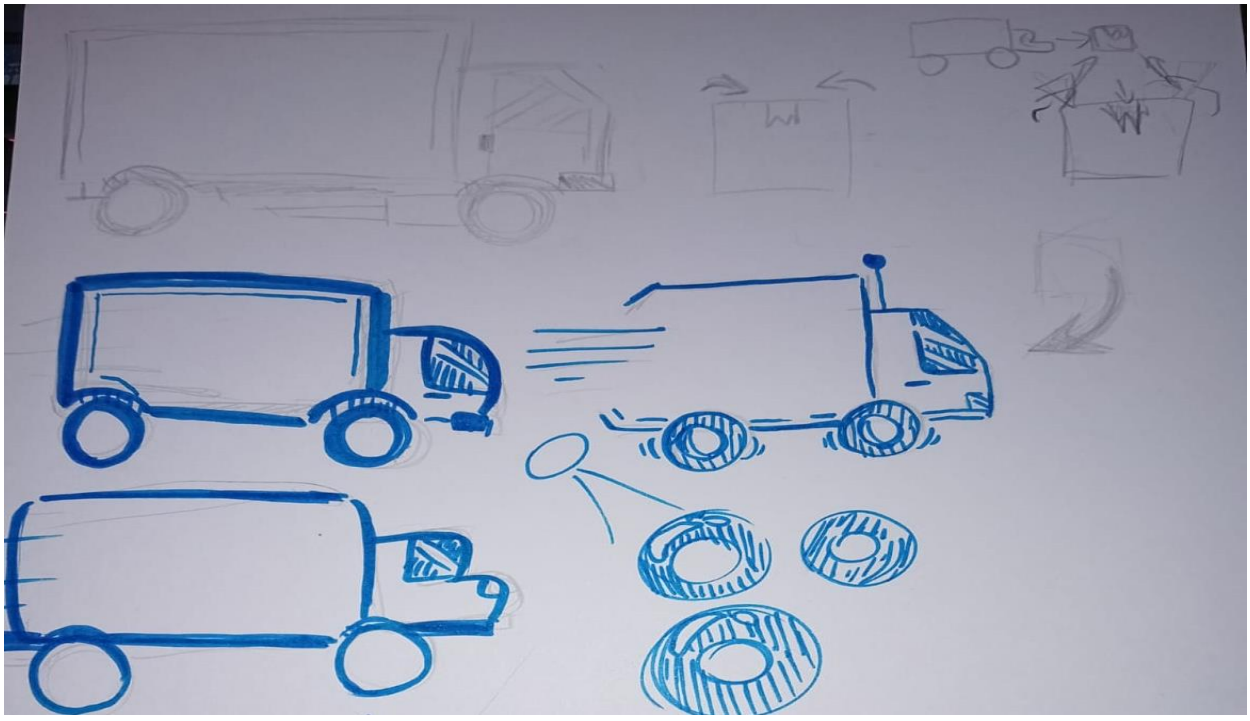


Figure 5 Logo Concept Not Definitive

3. PRE-CONSTRUCTION

Website Flow

- The website flow should be clearly defined, to ensure that users can place orders and manage them efficiently. A possible flow would be the following:
- **Home page:** Presents clear options for placing a new order or tracking an existing one.
- **Create a new order:** The user accesses a form where the details of the product, quantity, and delivery destinations are entered.
- **Order confirmation:** The system displays an order summary with the selected delivery options, and the user confirms the shipment.
- **Order tracking:** The user receives a tracking number that allows them to monitor the status of their delivery.
- **Viewing Past Orders:** Users can log in to view previous orders or reorder similar products.

Images, videos, logos, UX design

- For the construction phase, the necessary visual and design resources must be identified:
- **Images:** Photographs of delivery vehicles transported products and optimized routes.
- **Videos:** Instructions on how to use the site's features or testimonials from satisfied customers.
- **Logos and branding:** Ensure that the company's visual identity is consistent throughout the site.
- **UX design:** User experience design should be clear and accessible, especially in sections like the order form and tracking tools.

4. CONSTRUCTION Front-End and Back-End

Home Page

Figure 6 Home Page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="css/index.css">
    <title>Inicio</title>

</head>
<body>
    <div class="background-image"></div>
    <div class="login-register">
        <a href="/LOGIXPRESS/Login.php">Login/Register</a>
    </div>
    <section class="content-section">
        <header class="header-oval">
            <h1>Bienvenidos a LOGIXPRESS</h1>
            <p>Optimiza y ahorra en tus envíos de mercancía!</p>
        </header>
        <div class="container">
            <!-- Botón caja 1 - Inicio -->
            <a href="#" onclick="verificarLogin('Home.html')" class="box">
                <div class="tape"></div>
                <div class="icons">
                    <div class="icon up-icon">↑ ↑</div>
                    <div class="icon umbrella-icon">☂</div>
                </div>
                <p class="text">Inicio</p>
            </a>

            <!-- Botón caja 2 - Entrega -->
            <a href="#" onclick="verificarLogin('EntregaForm.php')" class="box">
                <div class="tape"></div>
                <div class="icons">
                    <div class="icon up-icon">↑ ↑</div>
                    <div class="icon umbrella-icon">☂</div>
                </div>
                <p>Entrega</p>
            </a>

            <!-- Botón caja 3 - Acerca De -->
            <a href="#" onclick="verificarLogin('Nosotros.php')" class="box">
                <div class="tape"></div>
                <div class="icons">
```

Figure 7 Code

This page was created using HTML, PHP, and JavaScript, along with some external and custom resources to structure and style a LOGIXPRESS platform login page.

PHP:

- HTML5: Provides the basic structure of the page, including header tags and navigation links, as well as a welcome section.

JavaScript:

- The checkLogin() function will allow access to other pages to be conditioned based on the user's authentication status. If the user is not logged in, they are automatically redirected to the login page.

CSS and Bootstrap:

- Uses Bootstrap 4.3.1 for layout styles, such as responsive formatting and the use of a container to organize content. A custom stylesheet (index.css) is also loaded to apply additional styling, such as the background image, header, and navigation box styling.

Google Fonts:

- Uses the Pacifico font to give text a distinctive style.

Navigation Structure:

- The page contains a series of interactive boxes that act as buttons to direct to different sections ("Home", "Delivery", "About Us" and "Help"). These boxes will only allow access if the user is logged in; otherwise, they are redirected to the login page.

Login Page

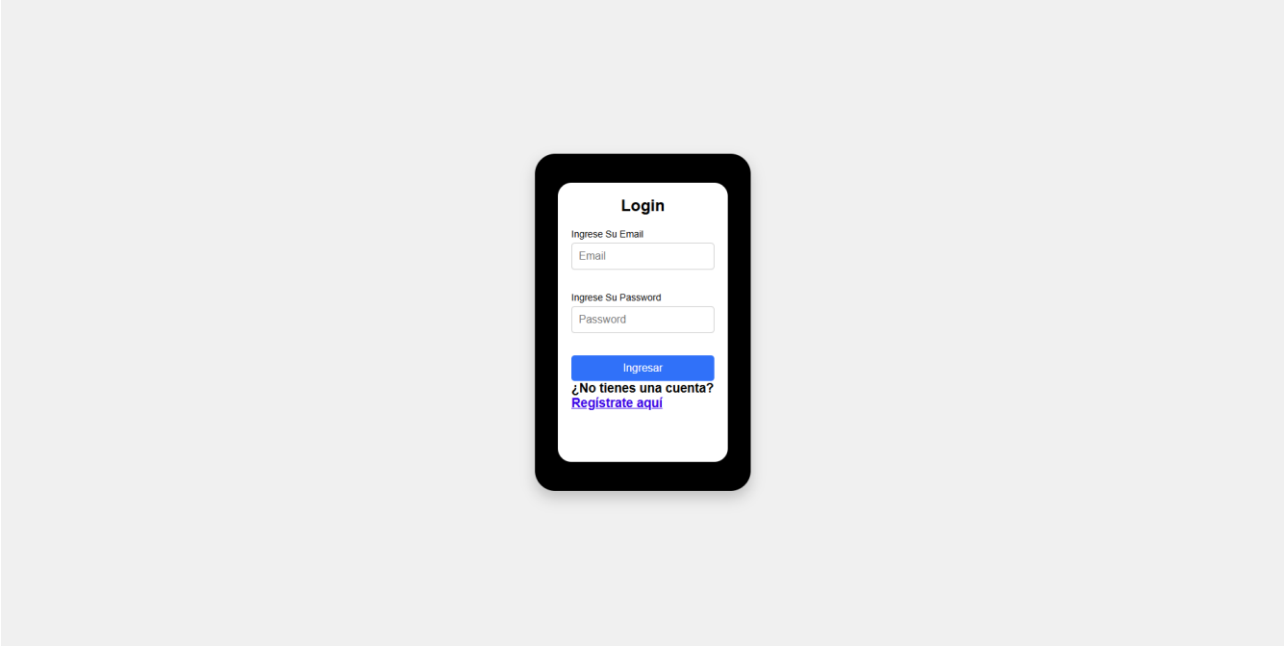


Figure 8 Login Page

```php
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/Login.css">
    <title>Login</title>
</head>
<body>
    <section class="tablet">
        <div class="screen">
            <form method="POST" action="">
                <h2>Login</h2>


                <?php if (isset($error)): ?>
                    <p style="color: red;"><?php echo $error; ?></p>
                <?php endif; ?>

                <label for="email">Ingrese Su Email</label>
                <input type="text" id="email" name="email" placeholder="Email" required>
                <br>
                <label for="password">Ingrese Su Password</label>
                <input type="password" id="password" name="password" placeholder="Password" required>
                <br>
                <button type="submit">Ingresar</button>
                <h3>¿No tienes una cuenta? <a href="/LOGIXPRESS/Register.php">Regístrate aquí</a></h3>
            </form>
        </div>
    </section>
</body>
</html>
```

Figure 9 Code Login Page

This page implements a login page using PHP, HTML and CSS for the LOGIXPRESS platform, allowing users to authenticate and access their account.

1. PHP and Sessions:

   Start a session with *session_start()* to manage the user's status and check if they are logged in. If the form is submitted *($_SERVER['REQUEST_METHOD'] === 'POST'),* the email and password entered by the user are captured.

2. Database Connection (PDO):

   The connection to the logixpress database is done through PDO (PHP Data Objects), which allows for a secure and configurable connection. PDO options include error handling with PDO::ERRMODE_EXCEPTION for robust management and PDO::FETCH_ASSOC to fetch the data as an associative array.

3. User Verification:

   A prepared query *($stmt = $pdo->prepare(...))* is used to look up the user in the database based on the email provided, avoiding SQL injection vulnerabilities. If the user is found and the password matches, the user ID is saved in the session *($_SESSION['user_id'])* and redirected to Home2.php. If the credentials do not match, an error message is displayed on the page.

4. HTML5 and Login Form:

   The login form has input fields for email and password, which are required. It uses a submit button to submit credentials via the POST method. It includes a link to register, in case the user does not have an account, redirecting them to the registration page.

5. CSS Styles:

   Applies a custom stylesheet (Login.css) to the visual layout of the login page, giving a particular style to the form and its container, simulating the appearance of a tablet for an attractive and professional visual look.

Register Page

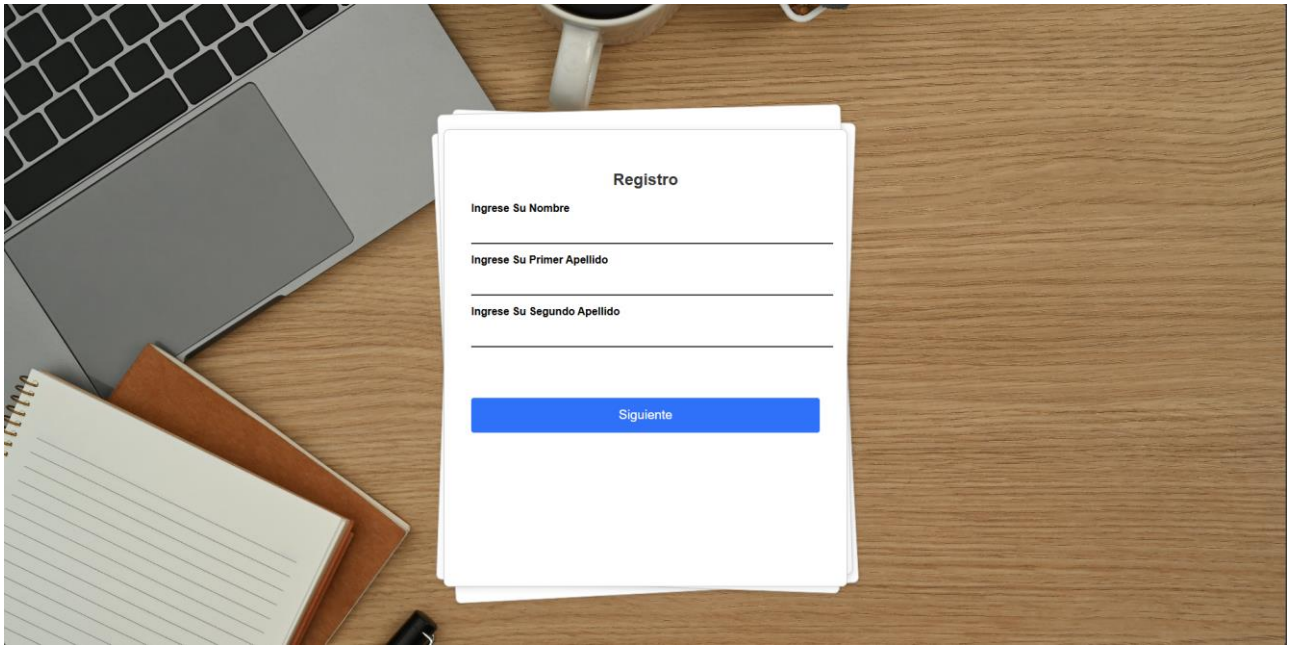


Figure 10 Register Page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/Register.css">
    <title>Register</title>
</head>
<body>

    <div class="background-image"></div>

    <section>
        <div class="stacked-paper-container">
            <div class="paper-layer" id="paper1"></div>
            <div class="paper-layer" id="paper2"></div>
            <div class="paper-layer" id="paper3"></div>

            <!-- Formulario 1 -->
            <div class="form-box form-page" id="form1">
                <form id="form1-content" method="POST">
                    <h2>Registro</h2>
                    <input type="hidden" name="form1" value="1">
                    <label for="nombre">Ingrese Su Nombre</label>
                    <input type="text" id="nombre" name="nombre" required>
                    <br>
                    <label for="primerApe">Ingrese Su Primer Apellido</label>
                    <input type="text" id="primerApe" name="primerApe" required>
                    <br>
                    <label for="segundoApe">Ingrese Su Segundo Apellido</label>
                    <input type="text" id="segundoApe" name="segundoApe" required>
                    <br>
                    <button type="button" id="nextBtn" onclick="guardarYMostrarFormulario2(event)">Siguiente</button>
                </form>
            </div>

            <!-- Formulario 2 (visible pero detrás del formulario 1) -->
            <div class="form-box form-page" id="form2" style="display: none;">
                <form id="form2-content" method="POST" action="Register.php">
                    <h2>Datos de la Empresa</h2>
                    <input type="hidden" name="form2" value="1">
                    <label for="empresa">Nombre De La Empresa</label>
                    <input type="text" id="empresa" name="empresa" required>
                    <br>
                    <label for="telefono">Ingrese el Teléfono de la Empresa</label>
                    <input type="tel" id="telefono" name="telefono" required>
                    <br>
                    <label for="email">Ingrese el Email de la Empresa</label>
                    <input type="email" id="email" name="email" required>
```

Figure 11 Code Register Page

```
        <!-- Formulario 2 (visible pero detrás del formulario 1) -->
        <div class="form-box form-page" id="form2" style="display: none;">
            <form id="form2-content" method="POST" action="Register.php">
                <h2>Datos de la Empresa</h2>
                <input type="hidden" name="form2" value="1">
                <label for="empresa">Nombre De La Empresa</label>
                <input type="text" id="empresa" name="empresa" required>
                <br>
                <label for="telefono">Ingrese el Teléfono de la Empresa</label>
                <input type="tel" id="telefono" name="telefono" required>
                <br>
                <label for="email">Ingrese el Email de la Empresa</label>
                <input type="email" id="email" name="email" required>
                <br>
                <label for="password">Password</label>
                <input type="password" id="password" name="password" required>
                <button type="submit">Finalizar</button>
            </form>
        </div>
    </div>
</section>

<script>
    function mostrarFormulario2() {
        document.getElementById('form1').style.display = 'none';
        document.getElementById('form2').style.display = 'block';
    }
</script>

<script src="./js/register.js"></script>

</body>
</html>
```

Figure 12 Code Register Page pt2

This page is designed for a registration page that uses a sequential forms format to collect user data in two steps, including personal information and company data.

1. General Structure:
   - The page starts with the standard *<!DOCTYPE html>* and sets the language to English.
   - In the header *(<head>),* the UTF-8 character encoding and viewport are set to ensure adaptability to mobile devices.
   - A custom stylesheet (Register.css) is included that defines the visual layout, as well as a page title.
2. Visual Layers and Background:
   - The background-image class sets a background image that fills the entire screen, adding an attractive visual context.
   - The "paper-layer" layers simulate a stacked folder effect in the interface, which organizes the forms in a visually structured way.
3. Sequential Registration Form:
   - The page contains two forms, each located in a box (form-box form-page), allowing data entry in two steps:
   - Form 1: Collects personal data (name, firstApe, secondApe). This form is initially displayed and contains a Next button that, when pressed, activates the next form.
   - Form 2: Collects business data (company, phone, email, password) and is initially hidden. It is only displayed when the user completes the first form and presses the Next button.
   - Each form includes required fields and labels to guide the user through data entry.

4. JavaScript:
    - The showForm2() function is a simple JavaScript function that changes the style of the forms, hiding the first one (form1) and showing the second one (form2), creating a transition between the two steps.

    - An additional JavaScript file (register.js) allows for additional functionality such as saving data to the session using AJAX and controlling transition logic between forms.

About Us Page

Figure 13 About Us Page

This page is designed for an informative “About Us” page for LOGIXPRESS, which presents the company’s mission, vision, and history, using a structured visual layout and a combination of text and images.

General Structure:

- The HTML page starts with the standard <!DOCTYPE html> and specifies the language as English.
- In the header (<head>), UTF-8 encoding and viewport are set to ensure that the page is responsive on mobile devices.
- A stylesheet (About Us.css) is loaded that provides the custom styles for the page.

Header with Logo:

- The page includes a header (<header>) with a header class and a logo-container section containing the LOGIXPRESS logo image. This reinforces the brand identity at the top of the page.
- The image (<img src="imagenes/LOGIXPRESS_LOGO_F2.png" alt="LOGIXPRESS Logo">) is used to display the company logo.

Informative Content Section:

- The main section of the page (<section>) includes a content box ( ) where the company's informative paragraphs are organized in a column format.
- Each information block ( ) combines text and a representative image:
- First Block: Explains the company's background, highlighting its focus on transportation efficiency and service quality.
- Second Block: Focuses on LOGIXPRESS' mission, its commitment to advanced technology and route optimization.
- Third Block: Describes the relationship of trust and experience that LOGIXPRESS has built in the sector.
- Images (<img src="imageX.png" alt="Descriptive image X">) are placed next to the text in each block to visually complement the information, alternating between the left and right side of the text in each section.

Navigation Button:

- At the end of the section, a navigation button ( ) is included that redirects users back to the platform's main page (Home2.php), making navigation easier.

Delivery Page

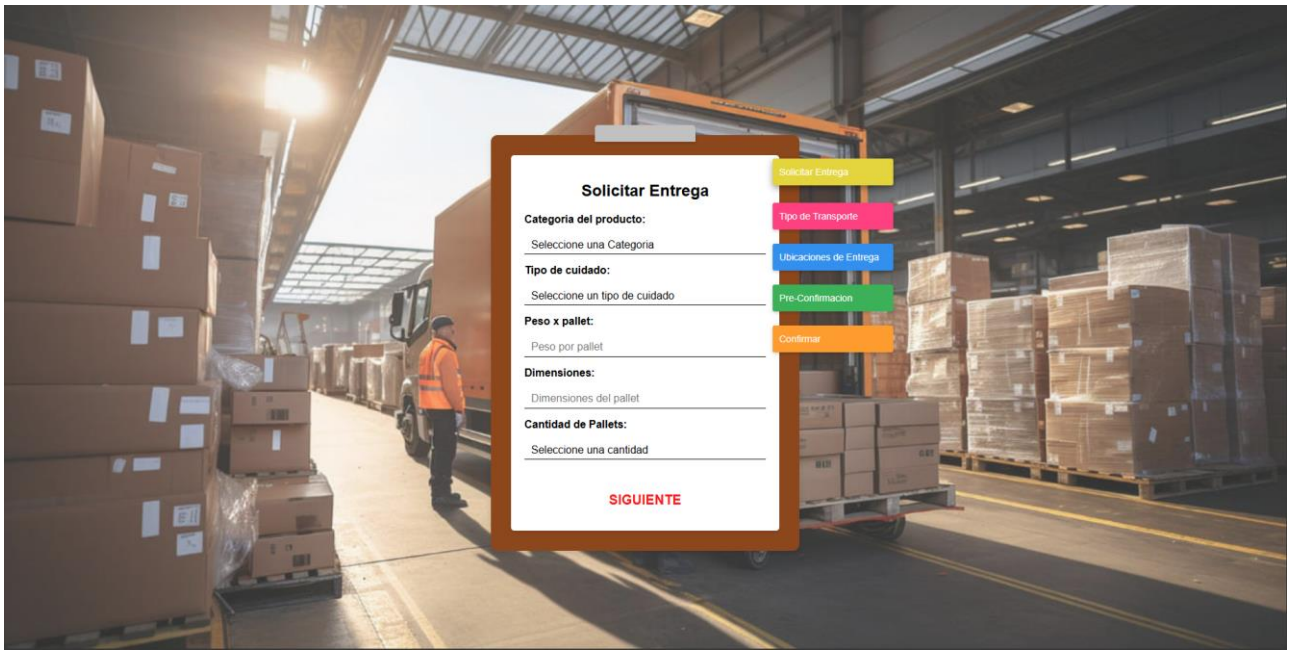


Figure 14 Delivery Reqeust Form Page

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/portapapeles.css">
    <title>Solicitar Entrega</title>
</head>
<body>

    <div class="background-image"></div>

    <div class="clipboard">
        <div class="paper">
            <!-- Contenido de Solicitar Entrega -->
            <div id="SolicitarEntrega" class="tab-content active-content">
                <form class="form" onsubmit="nextTab(event)">
                    <h2>Solicitar Entrega</h2>

                    <label for="producto">Categoria del producto:</label>
                    <select id="producto" name="producto" placeholder="Categoria del producto">
                        <option value="" disabled selected>Seleccione una Categoria</option>
                        <option value="" disabled selected>Seleccione una Categoria</option>
                        <option value="opcion1">Opción 1</option>
                        <option value="opcion2">Opción 2</option>
                        <option value="opcion3">Opción 3</option>
                    </select>

                    <label for="cuidado">Tipo de cuidado:</label>
                    <select type="text" id="cuidado" name="cuidado" placeholder="Tipo de cuidado">
                        <option value="" disabled selected>Seleccione un tipo de cuidado</option>
                        <option value="opcion1">Opción 1</option>
                        <option value="opcion2">Opción 2</option>
                        <option value="opcion3">Opción 3</option>
                    </select>

                    <label for="peso">Peso x pallet:</label>
                    <input type="text" id="peso" name="peso" placeholder="Peso por pallet">

                    <label for="dimensiones">Dimensiones:</label>
                    <input type="num" id="dimensiones" name="dimensiones" placeholder="Dimensiones del pallet">

                    <label for="cantidad">Cantidad de Pallets:</label>
                    <select type="num" id="cantidad" name="cantidad" placeholder="Cantidad de pallets">
                        <option value="" disabled selected>Seleccione una cantidad</option>
                        <option value="opcion1">Opción 1</option>
                        <option value="opcion2">Opción 2</option>
                        <option value="opcion3">Opción 3</option>
                    </select>
```

Figure 15 Code Delyvery Request Form

This page is designed for a delivery request page on the LOGIXPRESS platform. It uses a sticky note-inspired visual style and a tabbed structure to guide users through the different stages of a delivery request.

General Structure:

- The file starts with <!DOCTYPE html> and specifies the language as English.
- In the header (<head>), the UTF-8 character set and viewport are set to make the page responsive on mobile devices.
- A custom stylesheet (clipboard.css) is included that formats the elements, especially the sticky note style and the "paper" container where the forms are hosted.

Background Image and "Clipboard" Container:

- A background-image div adds a decorative background image to the page.
- Inside a clipboard div, the main content container (paper) is located, where the request form and the different tabs for each section of the request process are located.

Request Delivery Section:

- The first tab, RequestDelivery, contains a form that allows the user to specify details about the delivery:
- Form Fields: Fields such as “Product Category,” “Care Type,” “Weight x Pallet,” “Dimensions,” and “Pallet Quantity” are included with default labels and values to guide the user.
- Ship Button: A submit button with the class next-button allows the user to advance to the next step by submitting the data.

Tab Content:

- There are multiple tab-content sections that represent the different steps in the request flow:
    - Transport Type: Contains information or fields to select the type of transport.
    - Delivery Locations: Allows the user to specify locations.
    - Pre-Confirmation: A preview of the request before confirming.
    - Confirm: Presents the final confirmation of the request.
    - Only one section is visible at a time, controlled by the tab navigation JavaScript.

Tab Navigation with Sticky Notes:

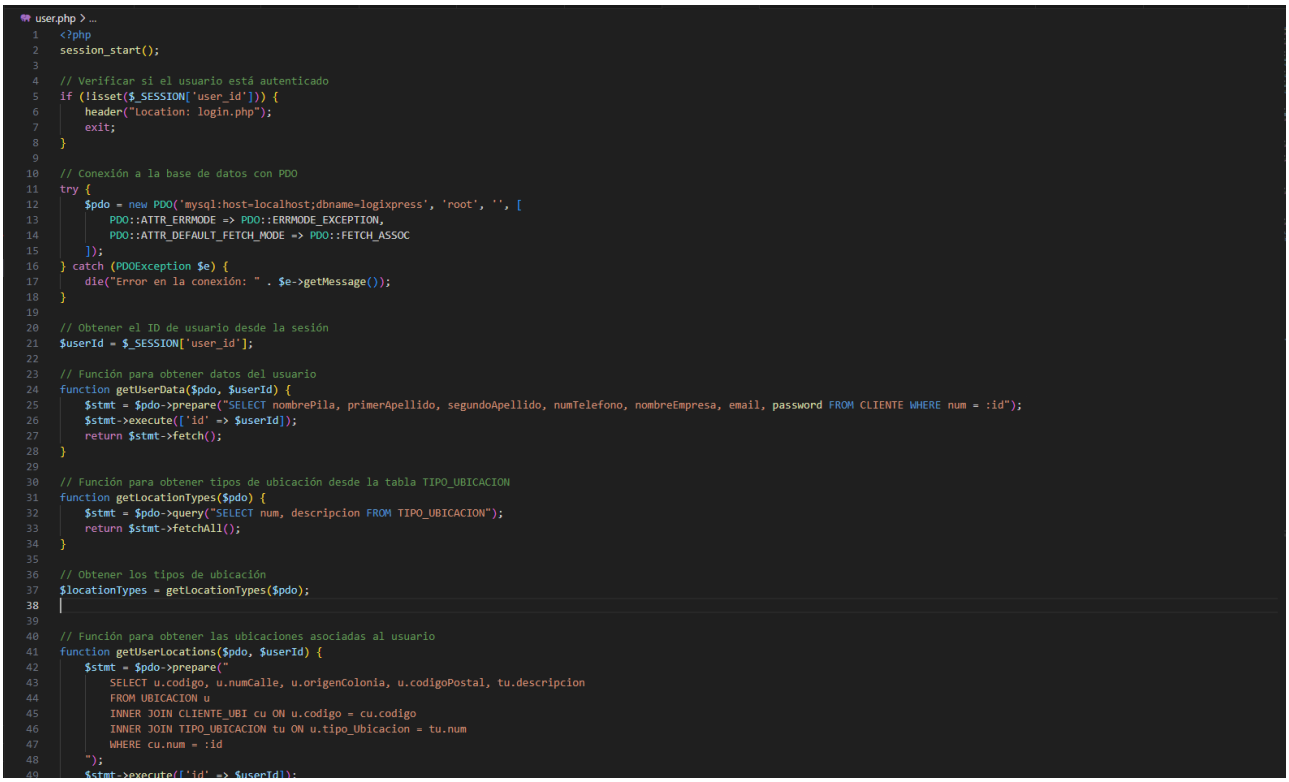
- Navigation between sections uses sticky-note buttons, each with a distinctive color and text indicating the corresponding step.
- Clicking on each button activates the corresponding section using the openTab() function defined in the post-it.js file.

JavaScript:

- The external JavaScript file (post-it.js) controls navigation between tabs using functions that toggle the content of each section on or off based on the button pressed.

User Account Administration Page

Figure 16 User Account Administration Page



user.php > ...

```php
<?php
session_start();

// Verificar si el usuario está autenticado
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit;
}

// Conexión a la base de datos con PDO
try {
    $pdo = new PDO('mysql:host=localhost;dbname=logixpress', 'root', '', [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);
} catch (PDOException $e) {
    die("Error en la conexión: " . $e->getMessage());
}

// Obtener el ID de usuario desde la sesión
$userId = $_SESSION['user_id'];

// Función para obtener datos del usuario
function getUserData($pdo, $userId) {
    $stmt = $pdo->prepare("SELECT nombrePila, primerApellido, segundoApellido, numTelefono, nombreEmpresa, email, password FROM CLIENTE WHERE num = :id");
    $stmt->execute(['id' => $userId]);
    return $stmt->fetch();
}

// Función para obtener tipos de ubicación desde la tabla TIPO_UBICACION
function getLocationTypes($pdo) {
    $stmt = $pdo->query("SELECT num, descripcion FROM TIPO_UBICACION");
    return $stmt->fetchAll();
}

// Obtener los tipos de ubicación
$locationTypes = getLocationTypes($pdo);

// Función para obtener las ubicaciones asociadas al usuario
function getUserLocations($pdo, $userId) {
    $stmt = $pdo->prepare("
        SELECT u.codigo, u.numCalle, u.origenColonia, u.codigoPostal, tu.descripcion
        FROM UBICACION u
        INNER JOIN CLIENTE_UBI cu ON u.codigo = cu.codigo
        INNER JOIN TIPO_UBICACION tu ON u.tipo_Ubicacion = tu.num
        WHERE cu.num = :id
    ");
    $stmt->execute(['id' => $userId]);
```

Figure 17 Code

This is an account administration page for the LOGIXPRESS platform, where users can manage their personal information, account details, and associated locations.

1. User Session and Authentication

- The page starts with session_start() to use the user's session.
- It verifies if the user is logged in by checking if $_SESSION['user_id'] is set; if not, it redirects to login.php to secure the page and prevent unauthorized access.

2. Database Connection

- It connects to the logixpress database using PDO (PHP Data Objects), which allows a secure connection and the configuration of options for error handling (ERRMODE_EXCEPTION) and the formatting of results (FETCH_ASSOC).

3. PHP Functions to Get User Data

- getUserData: Retrieves personal information of the user from the CUSTOMER table using the user_id of the session.
- getLocationTypes: Retrieves the location types from LOCATION_TYPE, which will be used in the location association form.
- getUserLocations: Retrieves the locations associated with the user from the LOCATION, CLIENT_UBI, and LOCATION_TYPE tables.

4. CRUD Operations in PHP

- Update Personal Information: Allows the user to update a specific field, validating that the field is allowed before running the update in the database.
- Delete Account: Deletes the user record in CLIENT, destroys the session, and redirects to login.php.
- Add New Location: Inserts a new location in LOCATION and associates it with the user in CLIENT_UBI.
- Update Location: Modifies the details of a specific location.
- Delete Location: Deletes the location from the CLIENT_UBI table.

5. HTML Interface

- Header (<head>): Sets the UTF-8 character encoding, the viewport to make the page responsive, and links the CSS file user.css for layout.

6. JavaScript for the Dynamic Interface

- Show Sections: The showSection() function allows you to toggle between sections of the interface, hiding the other sections.
- Edit Fields: The enableEdit, cancelEdit, and saveChanges functions control editing and saving user data dynamically, without requiring a page reload.

7. Vertical Navigation Structure

- The user navigation panel includes links to “Personal Information,” “Account Information,” and “Associated Locations” that allow users to navigate through the different sections of the page.

8. Section Contents

- Personal Information: Displays and allows editing of fields such as first name, last name, and phone number, with “Update,” “Confirm,” and “Cancel” buttons for each field.
- Account Information: Allows viewing and editing of the company name and email, as well as offering the option to delete the account.
- Associated Locations: Displays a table of locations and a form to add new locations. Users can update or delete specific locations.