

## Arquitectura de Información y Navegación (El "Cómo")

### A. Sitemap y Rutas

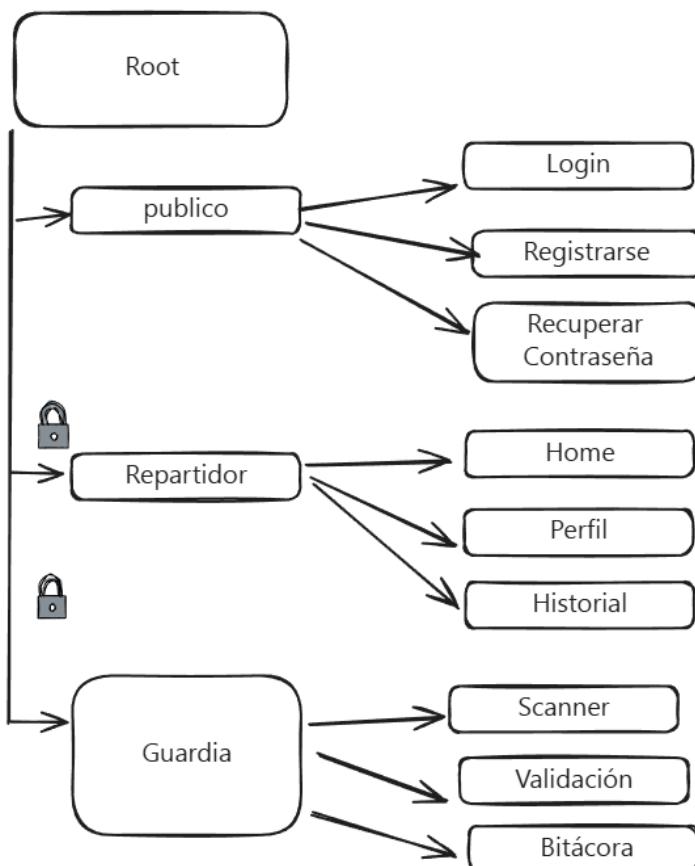
#### Rutas Públicas:

- `/` (Landing Page / Login)
- `/registro` (Formulario de alta: Datos + Fotos)
- `/recuperar-password`

#### Rutas Privadas (Requieren JWT):

- `/app/qr-pass` (Pantalla principal del Repartidor: Solo el QR y estado).
- `/app/perfil` (Edición de datos).
- `/admin/scanner` (Pantalla principal del Guardia: Cámara activa).
- `/admin/acceso/:id` (Pantalla de decisión: Foto gigante + Botón "Autorizar").

### Sitemap



## B. Accesibilidad y Navegación (A11y)

Para cumplir con "Navegación por teclado" y "Orden de tabulación".

### Orden de Tabulación Lógico (`tabindex`):

- El flujo debe ser: `Logo -> Navegación Principal -> Contenido Principal -> Acciones (Botones) -> Footer.`
- *Prohibido* usar `tabindex` mayor a 0. Usar el flujo natural del DOM.

### 2. Foco Visible:

- En CSS: `outline` nunca debe ser `none` sin un reemplazo visual claro. El usuario debe saber dónde está su "mouse invisible".

### 3. Semántica HTML (Skip Links):

- El primer elemento focusable de cada página debe ser un link oculto: "**Saltar al contenido principal**". Esto permite a los usuarios de teclado saltarse el menú en cada página.

### 4. Etiquetado ARIA:

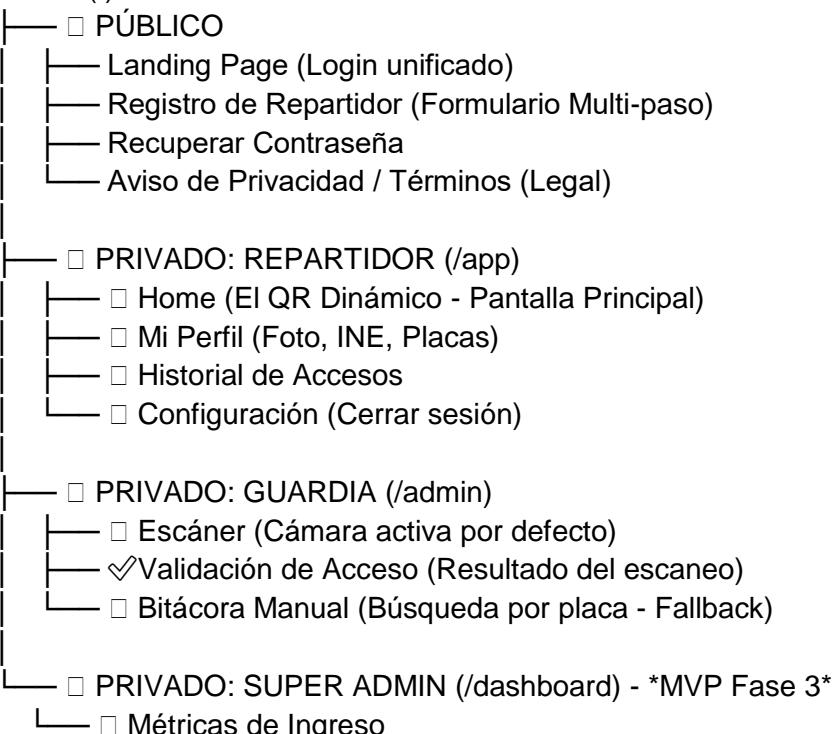
- Los botones que son iconos (ej. el botón de "Escanear") deben tener `aria-label="Escanear código QR"`.

## 1. Sitemap Completo (Arquitectura de Árbol)

Este mapa muestra la jerarquía de la información

Plaintext

ROOT (/)



## 2. Definición de Rutas (Para React Router)

Rutas Públicas (Acceso Libre)

Path (URL)	Componente	Descripción
/	LoginScreen	Entrada principal. Pide usuario/pass.
/register	RegisterDriver	Formulario de alta (Fotos + Datos).
/recovery	ForgotPassword	Recuperación vía email/SMS.
/privacy	PrivacyPolicy	Texto legal estático.

Rutas Privadas: Repartidor (Role: DRIVER)

Path (URL)	Componente	Descripción
/app/qr	DynamicQRView	<b>Vista Crítica.</b> Muestra el QR y contador de tiempo.
/app/profile	DriverProfile	Edición de datos (solo lecturas o updates).
/app/history	AccessHistory	Lista simple de "Entraste a Casa 4".

**Rutas Privadas: Guardia (Role: GUARD)**

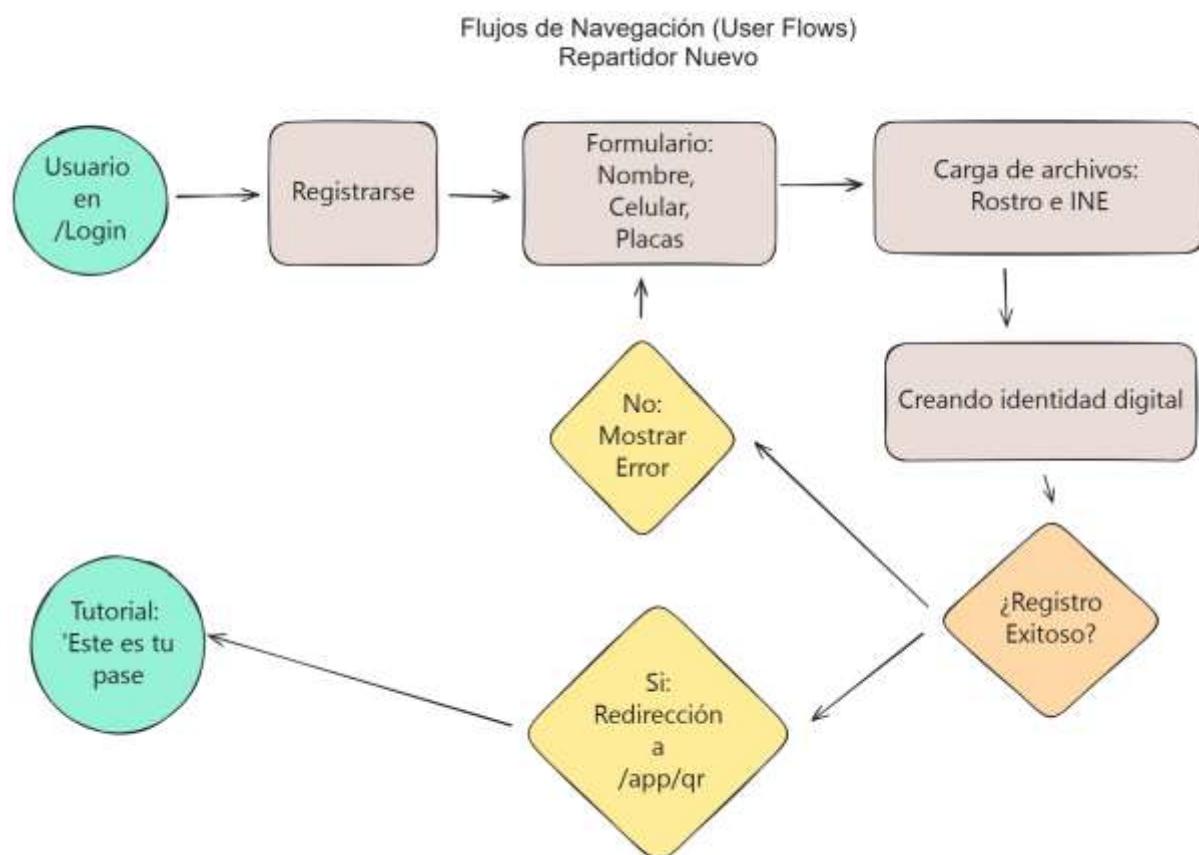
Path (URL)	Componente	Descripción
/guard/scanner	QRScannerView	Pantalla con la cámara abierta lista para leer.
/guard/validate/:id	AccessDecision	<b>Vista Crítica.</b> Muestra FOTO GIGANTE y botón AUTORIZAR.
/guard/logs	ShiftLogs	Resumen de quién ha entrado en su turno.

### 3. Flujos de Navegación (User Flows)

#### Flujo A: "El Repartidor Nuevo" (Onboarding)

**Meta:** Registrarse y obtener su primer QR.

1. **Inicio:** Usuario entra a [/](#). Clic en "¿No tienes cuenta? Regístrate".
2. **Datos:** Llena formulario (Nombre, Celular, Placas). Clic en "Siguiente".
3. **Fotos:** Sube Foto de Rostro y Foto de INE. Clic en "Finalizar".
4. **Carga:** Spinner de "Creando identidad digital...".
5. **Éxito:** Redirección automática a [/app/qr](#). Muestra Tutorial rápido ("Este es tu pase").

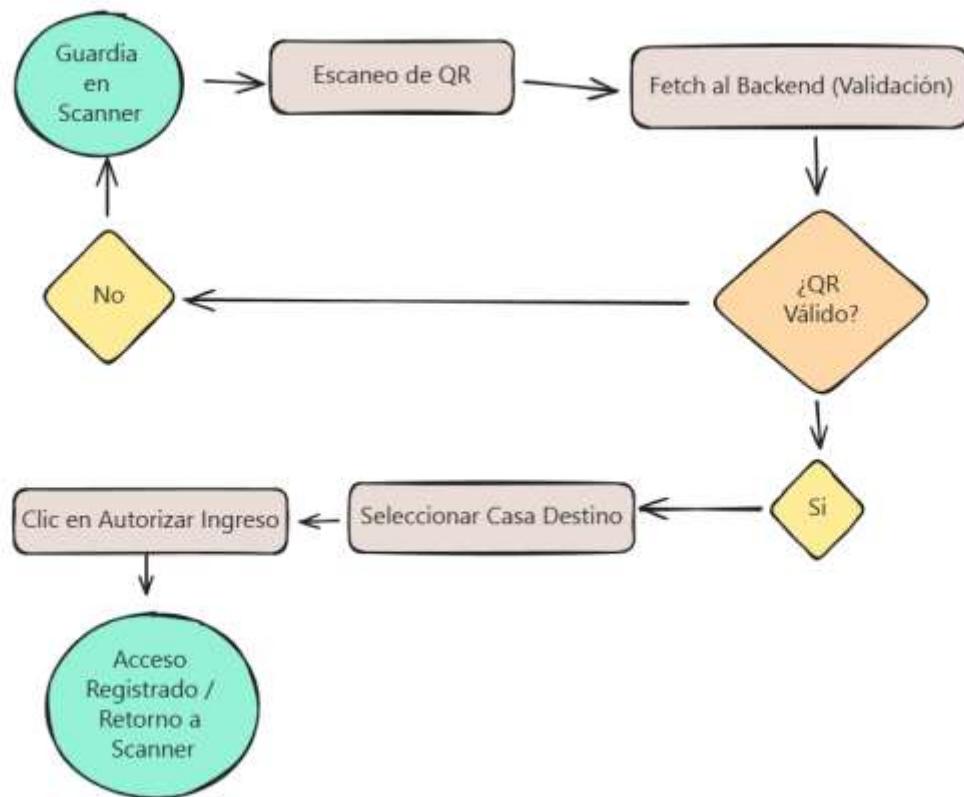


## Flujo B: "El Momento del Acceso" (Guardia)

**Meta:** Validar a un repartidor en menos de 10 segundos.

1. **Standby:** El guardia está en </guard/scanner>. La cámara está activa.
2. **Acción:** Repartidor muestra QR -> La cámara lo detecta automáticamente.
3. **Proceso:** El sistema hace fetch al backend.
4. **Decisión:** Redirección inmediata a [/guard/validate/{id\\_usuario}](/guard/validate/{id_usuario}).
  - *Muestra:* Foto Grande, Placas, Estado (ACTIVO/VENCIDO).
  - *Input:* Guardia pregunta "¿A dónde va?". Selecciona "Casa 45".
5. **Cierre:** Guardia presiona "**AUTORIZAR INGRESO**".
6. **Loop:** Mensaje de éxito ("Acceso Registrado") por 2 segundos -> **Auto-regreso a /guard/scanner** para el siguiente.

## Flujos de Navegación (User Flows) Acceso (Guardia)



## 4. Navegación por Teclado

Requisito de Accesibilidad (A11y):

El sistema debe ser operable sin mouse (por si la pantalla táctil de la tablet falla o se usa en PC de escritorio).

1. **Orden de Tabulación (Focus Order):**
  - En `/login`: Input Usuario -> Input Password -> Botón Entrar -> Link Registro.
  - En `/guard/validate`: Input Casa Destino -> Botón AUTORIZAR (El botón de cancelar debe ser accesible con Shift+Tab).
2. **Trampa de Foco (Focus Trap):**
  - Si abrimos un modal (ej. "¿Seguro que deseas salir?"), el tabulador NO debe poder seleccionar elementos del fondo.
3. **Feedback Visual:**
  - Todos los botones e inputs deben tener un estilo `:focus` claro (ej. borde azul grueso) para saber dónde está el cursor.

## Definición de Seguridad de Rutas y Navegación

### 1. Estrategia de Protección (Route Guards)

No basta con ocultar los botones en el menú. Debemos proteger la URL para que un repartidor "listo" no escriba `/admin/scanner` y trate de autorizarse a sí mismo.

Implementar un componente `<ProtectedRoute>` que verifique dos cosas:

1. **Autenticación:** ¿Existe un Token JWT válido en el LocalStorage?
2. **Autorización (Rol):** ¿El usuario tiene el rol correcto para ver esta página?

JavaScript

```
// Lógica esperada del componente ProtectedRoute (Pseudocódigo)
const ProtectedRoute = ({ children, requiredRole }) => {
  const { user, token } = useAuth(); // Hook personalizado

  // 1. Si no hay token, patada al Login
  if (!token) {
    return <Navigate to="/" replace />;
  }

  // 2. Si hay token pero el rol no coincide (ej. Repartidor queriendo entrar a Admin)
  if (requiredRole && user.role !== requiredRole) {
    return <Navigate to="/unauthorized" replace />; // O redirigir a su home
  }

  // 3. Pase usted
  return children;
};
```

## 2. Definición Detallada de Rutas

### □ Rutas Públicas (Public Routes)

*Accesibles para cualquier persona (anon).*

1. **/ (Login):**
  - **Propósito:** Punto único de entrada.
  - **Lógica:** Si ya estoy logueado, me redirige automáticamente a mi Home (`/app/qr` o `/admin/scanner`) según mi rol.
2. **/register (Registro Repartidor):**
  - **Propósito:** Formulario de alta.
  - **Lógica:** Al finalizar con éxito -> redirige a Login o auto-login.
3. **/unauthorized:**
  - **Propósito:** Pantalla de error "No tienes permiso para ver esto".

### Rutas Privadas (Protected Routes)

Rol: REPARTIDOR (Driver)

*Requiere role: 'driver'*

1. **/app/qr (Home):**
  - **Contenido:** El Código QR (TOTP) + Barra de tiempo + Foto de perfil pequeña.
  - **Seguridad:** El QR no es una imagen estática, es un canvas generado en vivo.
2. **/app/profile:**
  - **Contenido:** Datos personales. Botón "Cerrar Sesión".

Rol: GUARDIA (Guard)

*Requiere role: 'guard'*

1. **/admin/scanner:**
  - **Contenido:** Visor de cámara a pantalla completa.
  - **Lógica:** Al detectar QR válido -> Redirige a `/admin/validate/:id`.
2. **/admin/validate/:userId:**
  - **Contenido:** Ficha técnica del usuario escaneado.
  - **Seguridad:** Si el ID no existe o el token expiró, mostrar error.

### 3. Wireframes Textuales & Accesibilidad (A11y)

Aquí definimos **qué** va en la pantalla y **cómo** se navega usando solo el teclado (Tab / Enter).

#### A. Pantalla: Login (/)

**Objetivo:** Entrar rápido. **Diseño:** Centrado, limpio, logo arriba.

##### Estructura Visual:

1. Logo "Tag Human".
2. Título **<h1>**: "Bienvenido".
3. Input: Teléfono (type="tel").
4. Input: Contraseña (type="password").
5. Botón Primario: "Ingresar".
6. Link Secundario: "¿No tienes cuenta? Regístrate aquí".

##### □ Flujo de Teclado (Tab Order):

1. Foco inicial -> **Input Teléfono**.
2. Tab -> **Input Contraseña**.
3. Tab -> **Botón Ingresar** (Enter activa el submit).
4. Tab -> **Link Registro**.

#### B. Pantalla: Mi Pase (/app/qr) - [Vista Repartidor]

**Objetivo:** Mostrar el pase al guardia sin tocar nada. **Diseño:** Alto contraste. Fondo blanco para el QR, brillo al máximo.

##### Estructura Visual:

1. Header: Foto de perfil (Circular) + Nombre "Hola, Oscar".
2. **Área Principal (Main):**
  - Código QR (Grande, centrado).
  - Barra de progreso (Animación de 30s a 0s).
  - Texto **<h2>**: Placas "XYZ-123".
3. Footer: Botón "Mi Perfil" (Icono).

##### □ Flujo de Teclado:

1. Foco inicial -> **QR Container** (debe tener **aria-label="Código QR activo, actualizando cada 30 segundos"**).
2. Tab -> **Botón Perfil**.

## C. Pantalla: Validación ([/admin/validate/:id](#)) - [Vista Guardia]

**Objetivo:** Decisión rápida. Sí pasa o NO pasa. **Diseño:** Divide la pantalla. Izquierda: Foto. Derecha: Acciones.

### Estructura Visual:

1. **Columna Izquierda (Identidad):**
  - **FOTO GIGANTE** (Ocupa 40% pantalla). Vital para comparar con la cara real.
  - Nombre Completo.
  - Placas (Texto gigante).
2. **Columna Derecha (Acción):**
  - Input: "Número de Casa Destino" (Numérico).
  - Botón VERDE (Enorme): "AUTORIZAR ACCESO".
  - Botón ROJO (Pequeño/Texto): "Rechazar / Datos incorrectos".

### Flujo de Teclado (Crítico):

1. **Foco Inicial Automático** -> **Input Casa Destino**. (El guardia no debe tener que hacer clic, solo escribe el número).
2. Tab -> **Botón AUTORIZAR**.
3. Tab -> **Botón Rechazar**.