



**UNIVERSIDAD SANTO TOMÁS**  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
T U N J A

### **TÍTULO PROYECTO**

Dashboard para visualizar datos obtenidos de un sensor 4D/UHD+ TRUGRD STREAM de  
carretera

### **PROPONENTE(S)**

Ericka Julieth Sora Quintero  
1007226350  
2251178

### **DIRECTOR**

Ing. David Gerónimo Soler Gómez

Tunja

Fecha de presentación (15, 12, 2023)

## CONTENIDO

<b><u>1.</u></b>	<b><u>FICHA TÉCNICA DEL PROYECTO.....</u></b>	<b><u>3</u></b>
<b><u>2.</u></b>	<b><u>PLANTEAMIENTO DEL PROBLEMA .....</u></b>	<b><u>4</u></b>
<b><u>3.</u></b>	<b><u>JUSTIFICACIÓN.....</u></b>	<b><u>5</u></b>
<b><u>4.</u></b>	<b><u>OBJETIVOS.....</u></b>	<b><u>7</u></b>
<b><u>5.</u></b>	<b><u>MODELO DE DESARROLLO.....</u></b>	<b><u>8</u></b>
<b><u>6.</u></b>	<b><u>DESARROLLO DEL PROYECTO .....</u></b>	<b><u>12</u></b>
<b>6.1.</b>	<b>FASE 1. DEFINICIÓN DE REQUISITOS.....</b>	<b>12</b>
<b>6.2.</b>	<b>FASE 2. SELECCIÓN DE TECNOLOGÍA .....</b>	<b>14</b>
<b>6.3.</b>	<b>FASE 3. DISEÑO DEL SISTEMA .....</b>	<b>16</b>
<b>6.4.</b>	<b>FASE 4. DESARROLLO DE LA APLICACIÓN .....</b>	<b>19</b>
<b>6.5.</b>	<b>FASE 5. PRUEBAS.....</b>	<b>33</b>
<b><u>7.</u></b>	<b><u>CONCLUSIONES Y ANÁLISIS DE RESULTADOS .....</u></b>	<b><u>37</u></b>
<b><u>8.</u></b>	<b><u>REFERENCIAS.....</u></b>	<b><u>38</u></b>
<b><u>9.</u></b>	<b><u>ANEXOS .....</u></b>	<b><u>41</u></b>

## 1. FICHA TÉCNICA DEL PROYECTO

Título	<i>FrontEnd</i> de Dashboard para visualizar datos obtenidos de un sensor 4D/UHD+ TRUGRD STREAM de carretera
Autor (es)	Ericka Julieth Sora Quintero
Director del proyecto	David Gerónimo Soler Gómez
Palabras claves	Desarrollo de dashboard, sensor de carretera, React.
Descripción	
<p>Describa en forma corta y precisa (no más de 250 palabras) de qué trata el proyecto. Consiste en un solo párrafo, no contiene citas bibliográficas y se redacta en tiempo pasado. No debe contener siglas o abreviaturas (excepto aquéllas de uso generalizado).</p> <p>El proyecto consistió en la creación de un dashboard para un radar de carretera, utilizando inicialmente un mock de datos. Con el desarrollo de este proyecto se estableció una solución a la visualización de los datos traídos por un API; se presentó gráficamente a través de contadores y gráficos estadísticos, también de manera agradable y ordenada hacia el usuario, haciendo que se proporcionara información completa y actualizada. Se estableció un menú de navegación (<i>Inicio, sentido {sentido1, sentido2}, live tracking, carril, total de vehículos, tipo de vehículos {peatón, bicicleta, motocicleta, carro, camión corto, camión largo, indefinido}, tipo de estadísticas {volumen, velocidad, percentil 85, avance promedio, gap}, fecha de recuperación, histórico de objetos</i>) donde se resume la información que se puede encontrar dentro del dashboard, facilitando la visualización y evitando una mala experiencia de usuario; del mismo modo, se desarrolló una vista específica por cada ítem del menú, con la finalidad de evitar posible saturación de información en la vista principal.</p>	

## 2. PLANTEAMIENTO DEL PROBLEMA

La problemática a tratar en este proyecto radica en la falta de una alternativa web para la muestra gráfica, de una vasta cantidad de datos obtenidos de un sensor 4D/UHD+TRUGRD STREAM de carretera; el usuario necesita una solución en la cual se le muestren dichos datos de manera ordenada, precisa y actualizada, además de poder descargar reportes en formato Excel. Actualmente, para la visualización de datos (no gráficos) se cuenta con un software propiedad del fabricante del sensor 4D/UHD+TRUGRD STREAM de carretera; que tiene una licencia específicamente para el sistema operativo Windows.

El proyecto se centra en la construcción *FrontEnd* de un Dashboard para el despliegue de datos obtenidos del sensor 4D/UHD+ TRUGRD STREAM de carretera; éste debe ser interactivo, agradable e intuitivo al usuario. El acceso no debe ser restringido, por lo cual todos los usuarios tendrán permitida la entrada al dashboard.

### 3. JUSTIFICACIÓN

Ya hace varias décadas y hasta la actualidad, el uso e implementación de los Dashboard para la visualización de datos se han convertido en una pieza fundamental dentro de dicho proceso, esto se debe a que los Dashboard son una herramienta muy útil a la hora de la organización y análisis de información, traídos a través de conexiones a fuentes de datos. A continuación, los datos se consolidan en una vista ordenada y agradable al usuario, lo cual facilita la comprensión de los mismos de manera rápida y clara.

La importancia de un Dashboard con una vista ordenada e intuitiva es crucial para distintas finalidades, algunas de las cuales pueden ser: toma de decisiones, comprensión en temas de rendimiento, análisis, gestión de la información, entre otros. Al tener disponibles los datos obtenidos del 4D/UHD+ TRUGRD STREAM de carretera en diferentes formatos (gráfico y textual), se logra un mayor impacto, puesto que se proporciona información completa y actualizada.

Así mismo, el dashboard se presenta como la solución al problema de visualización de los datos obtenidos de un sensor 4D/UHD+ TRUGRD STREAM de carretera por parte de los usuarios; con el desarrollo de este proyecto, se permite que los usuarios tengan un acceso eficiente a dichos datos, creando una ruta de enlace gráfica, ordenada e interactiva y mejorando eficazmente la gestión de la información.

Se establece previamente un menú de navegación donde se resume la información que se puede encontrar dentro del dashboard, facilitando la visualización y evitando una mala experiencia de usuario; del mismo modo, se desarrolla una vista específica por cada ítem del menú, con la finalidad de evitar posible saturación de información en la vista principal. Para lograr una mayor cobertura de las necesidades de los usuarios, también se tienen en cuenta a aquellos que desean descargar reportes, por lo cual se crea la opción de descarga en formato Excel.

Del mismo modo, se debe tener en cuenta que para lograr una culminación más rápida y exitosa del proyecto, el cliente tomó la decisión de adquirir una plantilla llamada Crema pagando una licencia por su uso; esta plantilla nos serviría más adelante como base para la

construcción del proyecto tomando como guía de la misma gráficas y estilos, pero sin copiarlos directamente. Así mismo, cabe resaltar que los datos fueron suministrados directamente por el cliente a través de una serie de endpoint los cuales ya venían cargados con información, y el trabajo del scrum team solo fue utilizar los datos, pero sin manejarlos directamente en su lugar de origen (sensor).

Para finalizar, se recalca que todos los textos que se agreguen deben estar disponibles tanto en inglés como en español, como herramienta de ayuda para la globalización de la información.

## 4. OBJETIVOS

### 4.1. Objetivo General

Desarrollar un dashboard como alternativa web para la visualización de datos obtenidos de un sensor 4D/UHD+ TRUGRD STREAM de carretera, utilizando la metodología SCRUM.

### 4.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivo específico
1	Elaborar la especificación de requerimientos del <i>FrontEnd</i> del dashboard mediante diagramas de casos de uso, diagramas de clases y mockups, para definir funcionalidades y características solicitadas por el cliente.
2	Implementar el componente <i>FrontEnd</i> del dashboard que permita la visualización de estadísticas, información sobre detección, conteo de vehículos y control referente al sentido de movilización, basados en datos obtenidos por un sensor 4D/UHD+ TRUGRD STREAM de carretera, utilizando React js para su desarrollo.
3	Validar el comportamiento del dashboard mediante pruebas funcionales, verificando la actualización en tiempo real de la información y corroborando el correcto funcionamiento de todos los componentes.

Fuente: Autor

## 5. MODELO DE DESARROLLO

SCRUM se define como un marco de trabajo ágil, en el que se aplican constantemente un conjunto de buenas prácticas que facilitan en gran medida un buen trabajo en equipo. Con la metodología SCRUM se puede trabajar en cualquier tipo de proyecto sin importar que tan complejo sea, esto debido a que, es una herramienta a través de la cual los equipos se colaboran rápida y eficazmente, permitiendo realizar un trabajo de alto impacto.

SCRUM brindó varias ventajas durante el proceso de desarrollo del dashboard, que ayudaron a avanzar dentro del proyecto incrementalmente, haciendo especial énfasis en el control de tareas y metas. Se fundamentó principalmente en 3 conceptos: transparencia, inspección y adaptación; la transparencia hizo que el proceso tuviera mucha más visibilidad ayudando a detectar situaciones que pudieron afectar el resultado, la inspección ayudó a detectar eventos inesperados e indeseados durante el proceso, y la adaptación ayudó a hacer cambios o ajustes dependiendo de los eventos encontrados en la inspección (Navarro Cadavid, Andrés; Fernández Martínez, Juan Daniel; Morales Vélez, Jonathan, 2013, p.33). Inicialmente, se expusieron los roles de la metodología SCRUM que fueron imprescindibles durante el proceso de construcción del dashboard, estos roles son: Product Owner, SCRUM Master y SCRUM Team. Contrastando con el proyecto que se presentó, en este caso se exhibió quiénes fueron participantes activos dentro del proceso; como Product Owner se desempeñó el Project manager de la empresa Roch Electronics, el rol de SCRUM master lo desempeñó el supervisor asignado y team developer Ericka Sora

La metodología SCRUM consta de un ciclo de vida que permite agilizar los procesos eficientemente. Dicho ciclo inicia con la planeación del sprint, donde se deciden las tareas a realizar por cada uno de los miembros del equipo. Seguidamente se hace una reunión del equipo socializando lo que se hizo y lo que se quiere hacer. El refinamiento del backlog se utiliza para esclarecer cualquier duda que surja por parte del SCRUM Team developer (Javier Garzás, 2021). A continuación, se hace una revisión por parte del cliente en el trabajo realizado, en la cual el SCRUM master y el Product Owner estarán a cargo. Por último, se encuentra la retrospectiva del sprint, en la cual se identifican los procesos exitosos y los que no alcanzaron el estándar de calidad planeado. Dicho ciclo de vida de la metodología SCRUM, será implementado dentro del proceso de construcción del proyecto.

En paralelo con la metodología SCRUM, se hizo especial énfasis en cómo fue desarrollado el proyecto: En primera instancia, el proceso se inició con la obtención de datos desde una API suministrada por el cliente (estos datos fueron obtenidos de Endpoint suministrados por el cliente) que posteriormente fueron mostrados interactiva y gráficamente a través del Dashboard; adicionalmente en la vista principal del Dashboard se agregó un botón el cual tuvo la funcionalidad de exportar los datos actualizados y en tiempo real a una hoja de cálculo Excel. El Dashboard está complementado con un menú funcional previamente establecido donde sus ítems o sub-ítems están directamente enlazados a una interfaz propia, por lo cual se creó una vista conformada por diferentes graficas para cada uno de los ítems y sub-ítems del ya mencionado menú. Los ítems y sub-ítems del menú son los siguientes: *Inicio, sentido {sentido1, sentido2}, live tracking, carril, total de vehículos, tipo de vehículos {peatón, bicicleta, motocicleta, carro, camión corto, camión largo, indefinido}, tipo de estadísticas {volumen, velocidad, percentil 85, avance promedio, gap}, fecha de recuperación, histórico de objetos*. Así mismo las herramientas que inicialmente se utilizaron durante el proceso de construcción son: la plantilla crema implementada y configurada previamente para hacer la construcción del Dashboard sobre la misma, Docker compose, GitHub, la biblioteca React, Node js, Redux, Axios y Hojas de estilos Css.

Siguiendo con la idea anterior, se explica cuáles son las respectivas fases que se abarcaron dentro del proyecto; esto teniendo en cuenta que cada fase nombrada a continuación, pasó secuencial y ordenadamente por el ciclo de vida de dicha metodología.

**Fase 1:** Definir necesidades del cliente, especificando lo que desea obtener del proyecto, permitiendo puntualizar funciones y características que se requieran. Dentro de esta fase se llevan a cabo las siguientes actividades:

- **Actividad 1.1:** Definir requerimientos: Se realizó una reunión con el Product Owner, Scrum Master y Scrum Team, para definir el Product backlog del proyecto, así como un Sprint planning con el cual se abordó cada una de las tareas que surgieron
- **Actividad 1.2:** Desarrollo de CU: Los casos de uso se presentaron gráfica y textualmente, para proporcionar una estructura de funcionalidades y características expuestas por el Product Owner.

- **Actividad 1.3:** Elaboración de mockups: Con respecto a los mockups del dashboard, se llegó a un acuerdo con el cliente con la finalidad de optimizar tiempo y avanzar en desarrollo. Este acuerdo consistió en tomar como referencia la plantilla Crema, esto debido a que esta plantilla es usada durante la construcción del proyecto y la misma ya tiene un Demo con vista predefinida disponible.

**Fase 2:** Implementar componente *FrontEnd* utilizando datos traídos desde una API, incorporando dicha información gráficamente en la interfaz del sitio web a través de diferentes elementos, y permitiendo así una satisfactoria experiencia de usuario. Dentro de esta fase se llevan a cabo las siguientes actividades:

- **Actividad 2.1:** Montaje de información gráfica correspondiente al módulo de tipos de vehículos.
- **Actividad 2.2:** Montaje de información gráfica correspondiente al módulo de estadísticas.
- **Actividad 2.3:** Montaje de información gráfica correspondiente al módulo de otras características.
- **Actividad 2.4:** Implementación de la funcionalidad para exportar reportes.

**Fase 3:** Validar y corroborar que el funcionamiento de todos los componentes implementados dentro del sitio web sea satisfactorio. Dentro de esta fase se llevan a cabo las siguientes actividades:

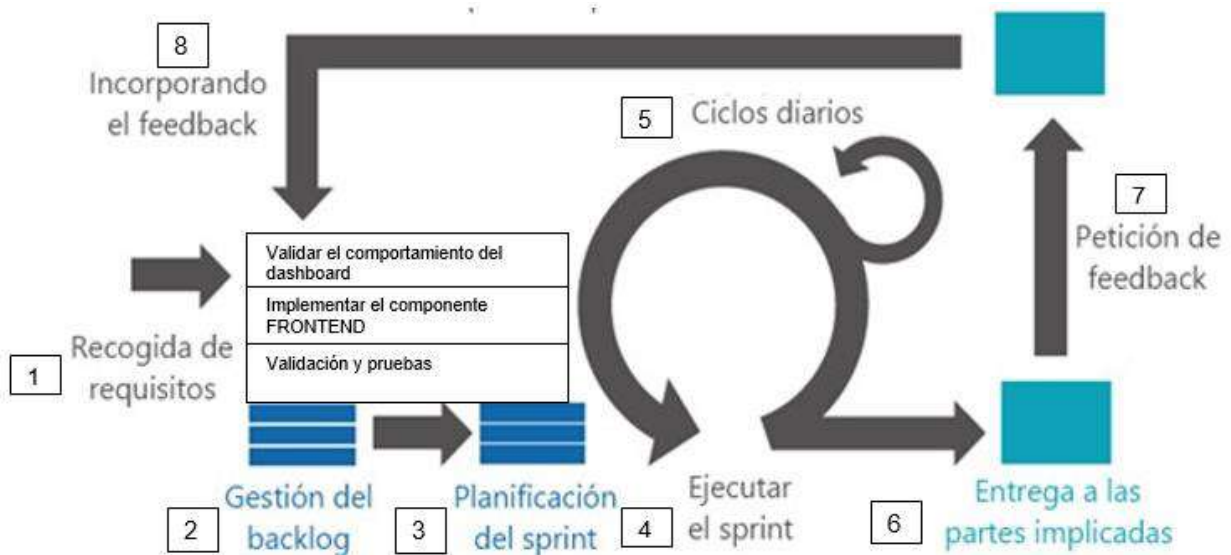
- **Actividad 3.1:** Diseñar un plan de pruebas
- **Actividad 3.2:** Verificación de la actualización en tiempo real de la información que se muestra en la interfaz: Una vez finalizó la fase de desarrollo del producto, se procedió a una fase de cierre y terminación, esta fase dio lugar al testing del dashboard por parte de Scrum team.
- **Actividad 3.3:** Verificación del funcionamiento de la funcionalidad exportar: Se hizo verificación y validación del correcto funcionamiento del botón exportar por parte de Scrum team.

Es muy importante tener en cuenta que el alcance de dicho proyecto finalizó al validar que el dashboard y el botón para exportar la información, fueron funcionales; Es decir, que la

información traída desde la API se visualizó gráficamente en las diferentes secciones del dashboard de manera clara y ordenada hacia el usuario; así mismo, la información coincidió con los datos obtenidos desde la API. Con lo anterior se da solución a la falta de una alternativa web con la cual visualizar datos obtenidos de un sensor 4D/UHD+ TRUGRD STREAM de carretera.

Finalmente, el objetivo al implementar la metodología SCRUM en este proyecto fue llevar a cabo y concluir todas las tareas propuestas en cada uno de los sprint, esto se debe lograr de manera planificada y colaborativa, para así, culminar satisfactoriamente el resultado propuesto inicialmente. Este proceso se llevó a cabo de manera conjunta y en equipo, teniendo en cuenta que cada uno de los miembros del SCRUM team fue una pieza importante para la finalización satisfactoria del proyecto.

Figura 1. Metodología SCRUM



Fuente: adaptada de:

<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

En la Figura1. se observa el ciclo de vida de la metodología Scrum, en la misma se ha identificado inicialmente la recogida de requisitos donde se plasman los 3 pilares generales que se abordan en el proyecto, los cuales son: Validar el comportamiento del dashboard, implementar el componente *FrontEnd* y pruebas.

## 6. DESARROLLO DEL PROYECTO

A continuación, se definen las fases con las cuales cuenta el proyecto de software.

### 6.1. Fase 1. Definición de requisitos

En esta fase se definen las necesidades del cliente, especificando lo que desea obtener del proyecto, permitiendo puntualizar funciones y características que se requieran. Para desarrollar el levantamiento de requerimientos se realizaron varias reuniones con Product Owner, SCRUM y Scrum team. Los siguientes son los requerimientos que se definieron:

- El dashboard debe ser intuitivo y agradable al usuario.
- El dashboard debe cumplir con la construcción de los siguientes módulos:
  - Módulo correspondiente a Tipos de vehículos.
  - Módulo correspondiente a Tipos de estadísticas.
  - Módulos correspondientes a Home, Dirección, live tracking, carriles,
  - fecha de recuperación e histórico de objetos
- Implementación del botón exportar y traducciones.

Con las funcionalidades y características plenamente identificadas, se procede a la elaboración del backlog, en el cual se listaron todas las tareas que se realizaron durante el desarrollo del proyecto. A continuación, se muestra el backlog:

Tabla 2. Backlog

Sprint	Descripción
<b>1</b>	Reunión para definir requerimientos
<b>2</b>	Desarrollo de casos de uso
<b>3</b>	Diagrama de clase
<b>4</b>	Diseño de mockups
<b>5</b>	Reunión de validación
<b>6</b>	Elección de herramientas
<b>7</b>	Aprobación de herramientas
<b>8</b>	Adecuación herramientas para desarrollo <i>FrontEnd</i>
	Instalación de librerías
<b>9</b>	Montaje de información gráfica correspondiente al módulo de tipos de vehículos
<b>10</b>	Montaje de información gráfica correspondiente al módulo de estadísticas
<b>11</b>	Montaje de información gráfica correspondiente a los Módulos de Home, Dirección, live tracking, carriles, fecha de recuperación e histórico de objetos.
<b>12</b>	Implementación del botón exportar
<b>13</b>	Verificación del funcionamiento del botón exportar

Fuente: Autor

## 6.2. Fase 2. Selección de tecnología

Se debe presentar la descripción de la tecnología utilizada para el desarrollo de software, paradigma de desarrollo, lenguaje, base de datos, entre otros aspectos.

Tabla 2. Fase 2

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> <li>• Plantilla Crema</li> <li>• Axios</li> <li>• Redux</li> </ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"> <li>• Docker compose</li> <li>• Visual Studio Code</li> <li>• React js</li> <li>• Hojas de estilos</li> <li>• Node js</li> </ul>
Salidas	Dashboard final
Tiempo	6 meses

Fuente: Autor

**Plantilla Crema:** Crema es una plantilla basada en React js, que se utiliza como punto de inicio para comenzar un próximo proyecto, y que ya contiene un kit de interfaz de usuario que proporciona una base de código. Un punto importante es el hecho de que admite la biblioteca y el kit de herramientas de Redux, “Crema es una plantilla de administración basada en React con muchas funciones que está configurada con las dos bibliotecas de interfaz de usuario principales (MUI-5, Ant Design-5) y bibliotecas y tecnologías de tendencia”. (Crema CRA, 2023).

Así mismo no es una plantilla que se pueda adquirir de manera gratuita, es una plantilla de paga que cuesta

\$29 dólares por una Licencia: Regular con 6 meses de soporte

\$900 dólares por una Licencia: Soporte extendido con 6 meses de soporte

**Axios:** Es un paquete que nos permite hacer peticiones o llamadas a los datos de un enlace HTTP (axios, 2023).

**Redux:** Es una librería de código abierto de JavaScript que ayuda con la gestión del estado en aplicaciones. Es normalmente usada con librerías como React y Angular Para la construcción de interfaces (redux, 2023).

**Docker compose:** Herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores (Docker docs, 2023)

**Visual Studio Code:** Es un editor de código fuente eficaz, que admite trabajar con diferentes lenguajes debido a las extensiones con las que cuenta (visual Studio Code Docs, 2023).

**React Js:** Es una librería robusta JavaScript de código abierto utilizada para la creación de interfaces y aplicaciones web dinámicas, con el objetivo de facilitar el desarrollo de las mismas (React Dev, 2023)

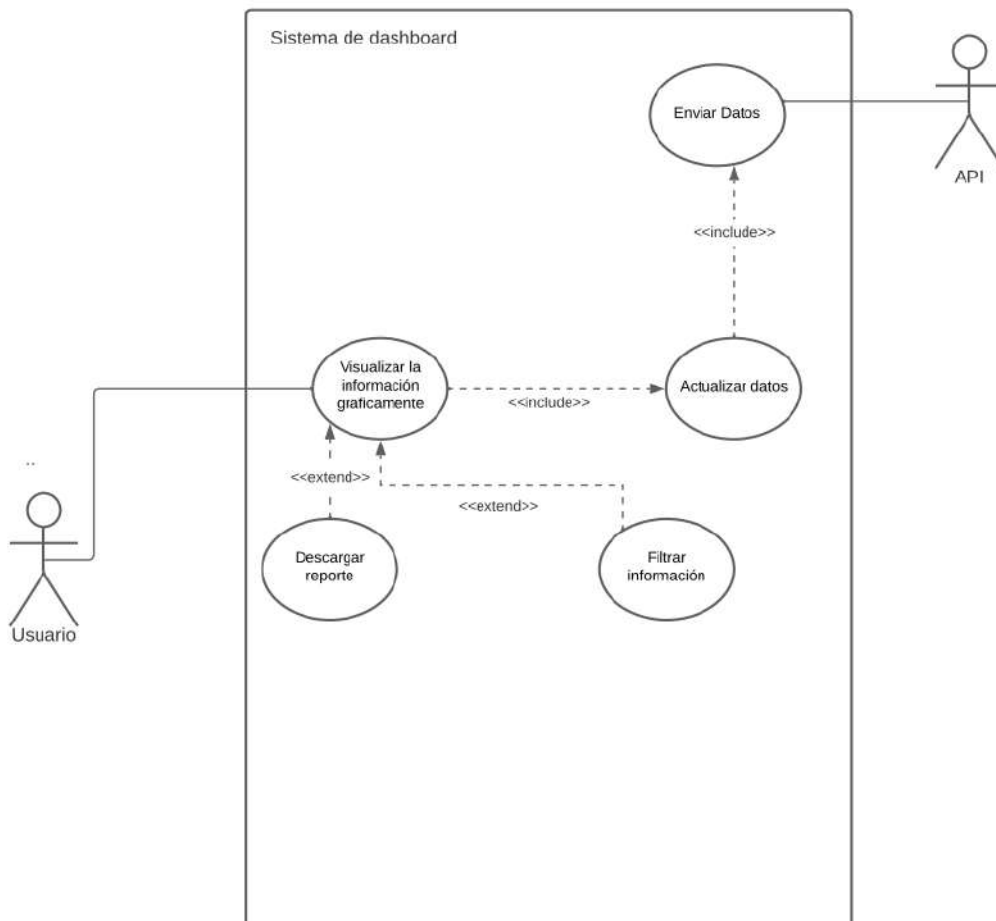
**Hojas de estilos:** Es un archivo de extensión .css que indica el formato y aspecto que tomara la información en internet (Mozilla, 2023).

**NODE JS:** Es un entorno en tiempo de ejecución de un solo hilo para la capa del servidor (About Node.Js, 2023).

### 6.3. Fase 3. Diseño del sistema

En esta sección se deben incluir los diagramas que representan el diseño del software, por ejemplo: casos de uso, entidad relación, clases y demás que junto con el director consideren pertinentes. Se procedió a la creación de los casos de uso como estructura inicial para expresar los requisitos:

Figura 2. Caso de Uso

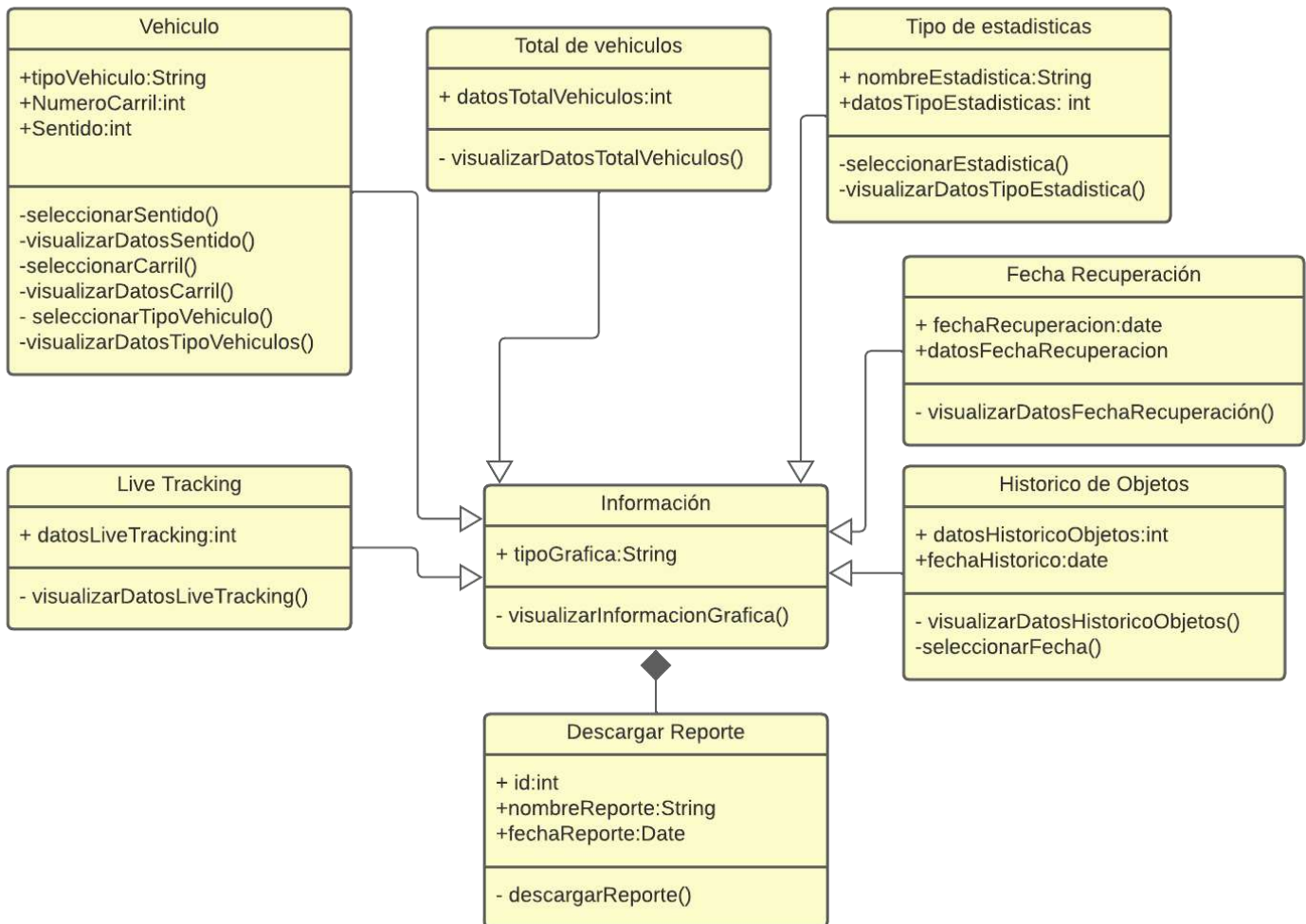


Fuente: Autor

Como requisito principal, se pide por parte del cliente la visualización de información gráfica en el dashboard, no especifica tipos gráficas, pero si se especifica que sea una vista ordenada y agradable. Para visualizar la información en el dashboard se deben actualizar datos, y para actualizar datos primeramente se debe hacer el envío de los mismos desde la API. Opcionalmente, para hacer la visualización de la información se puede descargar un reporte o filtrar la información del dashboard.

De igual manera se creó el diagrama de clases para mostrar las relaciones entre los elementos:

Figura 3. Diagrama de clases

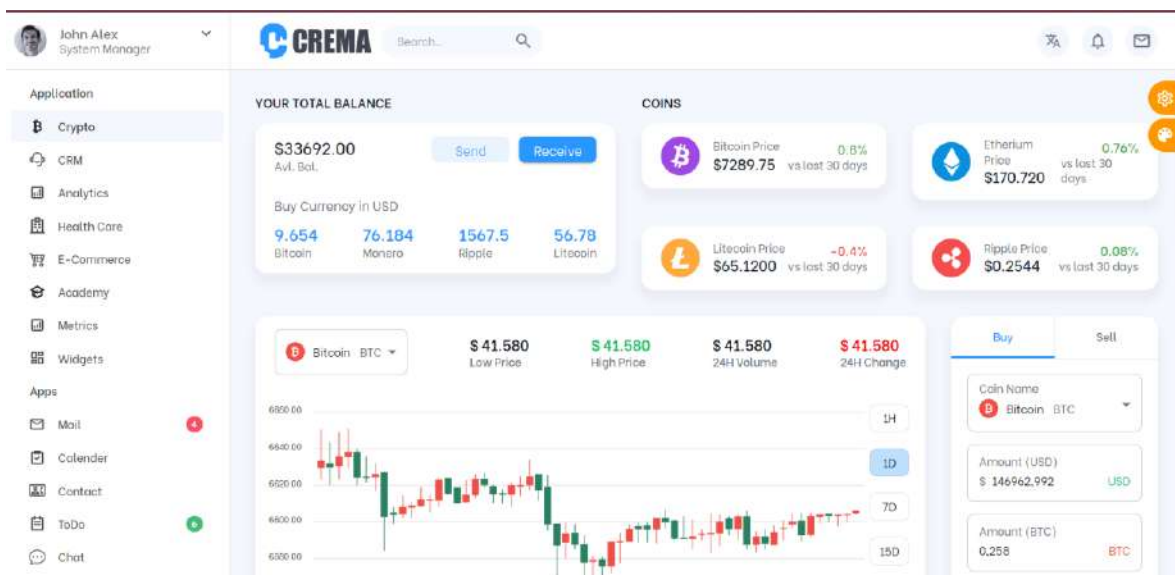


Fuente: Autor

Se toma como clase principal la clase información, de esta clase se heredan atributos a las clases secundarias las cuales son: live tracking, vehículo, total de vehículos, tipo de estadísticas, fecha de recuperación, histórico de objetos. Se tiene una relación de composición de la clase información como clase “contenedor” y la clase “descargar reporte” como la clase contenida, por lo cual, si la clase “información” desaparece, la clase “descargar reporte” también desaparece.

Con respecto a los mockups del dashboard, se tomó como referencia la plantilla Crema, esto debido a que esta plantilla es usada durante la construcción del proyecto y la misma ya tiene un Demo disponible. Esta es una plantilla de paga que necesita ciertas licencias para poderlas trabajar. En un inicio el cliente decide que para optimizar tiempo sería mejor trabajar sobre una plantilla así se podría utilizar ciertos recursos que se brindaban en la misma. El Demo nos da una vista predefinida en la cual nos podemos inspirar y así optimizar el tiempo para no hacer un doble trabajo; además se debe tener en cuenta que se debe mantener el tema original de la plantilla.

Figura 4. Plantilla Crema



Fuente: Autor

Con los items anteriormente mencionados, se concluye que se abarcó en su totalidad el objetivo1 propuesto para este proyecto.

#### 6.4. Fase 4. Desarrollo de la aplicación

Describir la estructura lógica del proyecto.

Primeramente, se debe hacer una comparación y separación entre lo que ya había predefinido en la plantilla que se reutilizó, y lo que se desarrollo desde cero y se implementó. Se desarrollaron un total de 31 paginas distintas, las cuales fueron desarrolladas en total autonomía. En ninguno de los casos, ninguna de las 31 páginas echas por Scrum Team tomo como base una pagina entera que se obtuviera de la plantilla y la editó.

1. Todas las gráficas que aparecen en el dashboard de tipo: rectangulares, de área, de líneas y tablas fueron reutilizadas de componentes que había en el dashboard, teniendo en cuenta que estas graficas solo funcionaban con datos locales. A continuación, una muestra de la gráfica de área antes

```
import React from 'react';
import {
  Area,
  AreaChart,
  CartesianGrid,
  ResponsiveContainer,
  Tooltip,
  XAxis,
  YAxis,
} from 'recharts';
import data from './data';
import { useThemeContext } from '@crema/context/AppContextProvide

const SimpleAreaChart = () => {
  const { theme } = useThemeContext();
  return (
    <ResponsiveContainer width='100%' height={200}>
      <AreaChart
        data={data}
        margin={{ top: 10, right: 0, left: -25, bottom: 0 }}
      >
        <XAxis dataKey='name' />
        <YAxis />
        <CartesianGrid strokeDasharray='3 3' />
        <Tooltip />
        <Area type='monotone' dataKey='uv' fill={theme.palette.pr
      </AreaChart>
    </ResponsiveContainer>
  );
};

export default SimpleAreaChart;
```

El equipo de Scrum Team llegó a adecuar dichas graficas para que funcionaran con datos traídos de endpoint con axios y redux, también teniendo en cuenta gestiones de estado para que los datos no se alteraran entre si. Gráfica de área después

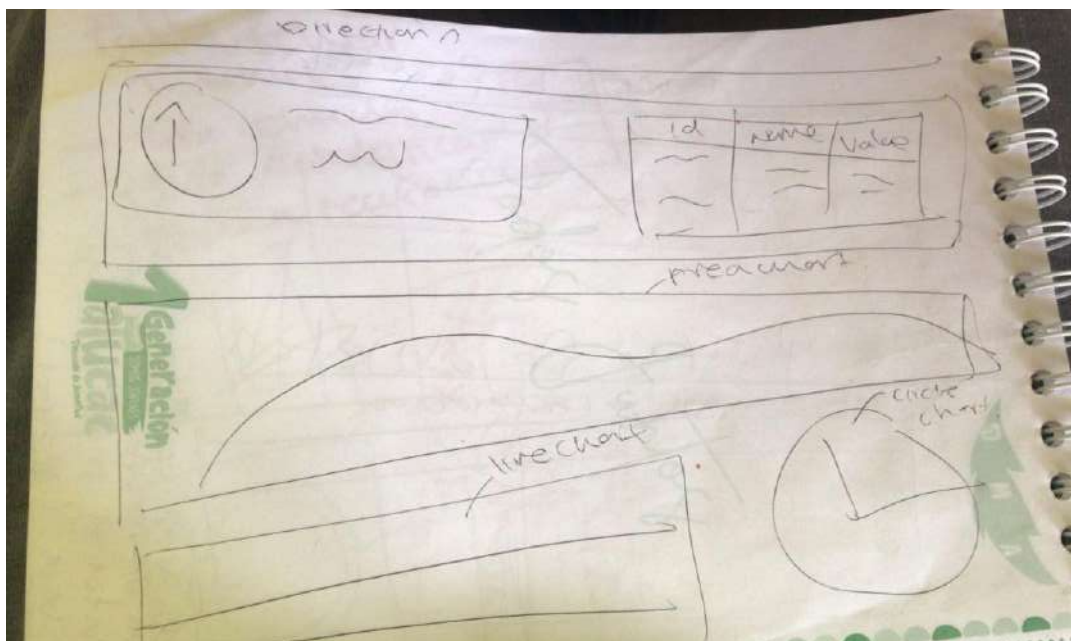
```
art-micro-mb > frontend > src > pages > extraPages > Components > Direction2 > JS AreaChart.js > ...
1 import React, { useEffect } from 'react';
2 import {
3   Area,
4   AreaChart,
5   CartesianGrid,
6   ResponsiveContainer,
7   Tooltip,
8   XAxis,
9   YAxis,
10 } from 'recharts';
11
12 import { getUsers5 } from 'redux/actions';
13 import { useDispatch, useSelector } from 'react-redux';
14
15 export default function AreaChartDirection() {
16   const dispatch = useDispatch();
17
18   const reducerData = useSelector((state) => state.todo);
19   const { graphic5 } = reducerData;
20   console.log(graphic5);
21
22   useEffect(() => {
23     dispatch(getUsers5());
24   }, [dispatch]);
25
26   console.log(graphic5);
27   return (
28     <ResponsiveContainer width={600} height={270}>
29       <AreaChart data={graphic5} margin={{top: 10, right: 0, left: -25, bottom: 0}}>
30         <XAxis dataKey='name' />
31         <YAxis />
32         <CartesianGrid strokeDasharray='3 3' />
33         <Tooltip />
34         <Area type='monotone' dataKey='value' fillOpacity='0.3' fill='#f59821' />
35       </AreaChart>
36     </ResponsiveContainer>
37   );
38 };
39
```

2. Los estilos tales como colores de fondo, tamaños y tipos de letra, entre otros, que aparecen en el dashboard son utilizados como referencia para lucir una apariencia

similar, pero cabe aclarar que estos se hicieron explícitamente por equipo de Scrum Team, no fueron copiados u obtenidos directamente de la plantilla.

3. La estructura de cada una de las 31 páginas que se crearon para la muestra de datos, fueron hechas explícitamente por el equipo de Scrum Team teniendo en cuenta que las mismas en un inicio estaban vacías. Este proceso en primera se hizo por medio de sketch que fueron inspirados visualmente por algunos elementos de la plantilla para dar estructura básica, que después se plasmo en los archivos de estructura específicos en cada página. Estos archivos de estructura no fueron copiados ni inspirados de ningún elemento de la plantilla, el orden en el que se colocaron los elementos y la selección de columnas filas o tarjetas fue totalmente autónomo.

Ejemplo de uno de los sketch de la pagina Direccion



A continuación, una muestra del archivo de estructura de la página Home.

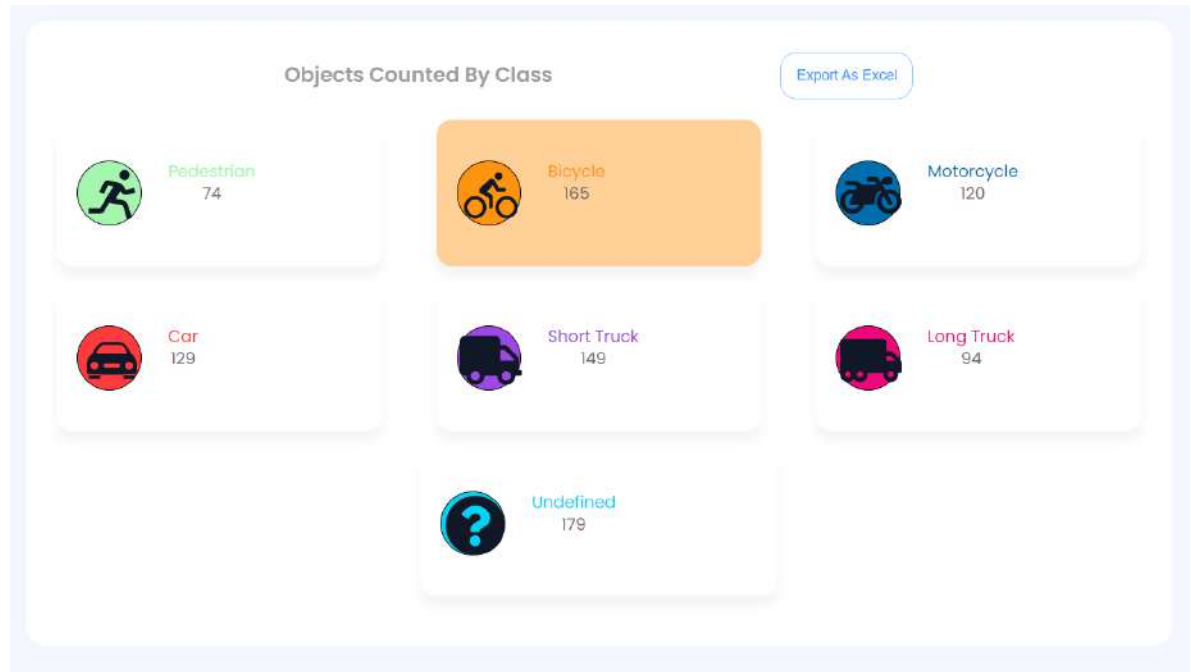
```
smart-micro-mb > frontend > src > pages > extraPages > Components > Home > JS StructureByClassjs > StructureByClass
1 import React, { useEffect } from "react";
2 import BotonExcel from "./BotonExcel";
3 import { useDispatch } from "react-redux";
4 import { getDates } from "redux/actions";
5 import './structureByClass.css';
6 import { BiRun } from "react-icons/bi";
7 import { IoIosBicycle, IoIosCar } from "react-icons/io";
8 import { FaMotorcycle, FaQuestionCircle, FaTruckMoving, FaTruck } from 'react-icons/fa';
9 import { AiOutlineArrowUp, AiOutlineArrowDown } from "react-icons/ai";
10 import { AiFillSignal, AiOutlineDashboard, AiOutlineInfoCircle, AiOutlineToTop, AiOutlineVerticalAlignMiddle } from "react-i
11 import IntlMessages from "@crema/utility/IntlMessages";
12 import CircularProgressHome from "./CircularProgress";
13 import AreaChartHome from "./AreaChart";
14 import ChartRectanglesHome from "./ChartRectangles";
15 import TotalStatistics from "./Total";
16 import { DataLineChartByStatistics } from "../DataLineChart";
17 import { DatacountDirection } from "../DataCount";
18
19
20 function StructureByClass({ count, description }) {
21   const dispatch = useDispatch();
22
23   useEffect(() => {
24     dispatch(getDates());
25   }, [dispatch])
26
27   return (
28     <>
29     <div className="white-container-center">
30       <div className="file-container-button">
31         <p className="fontTitleButton"><IntlMessages id='pages.extraPages.statistics.countedbyclass' /> </p>
32         <div className="button-excel"> <BotonExcel /> </div>
33       </div>
34       <div className="file-container">
35         <div className="card_block1 cardh1">
36           <div className="card-icon "><BiRun /></div>
37           <div className="card-font">
38             <div className="font-description">{description}</div>
39             <div>{count}</div>

```

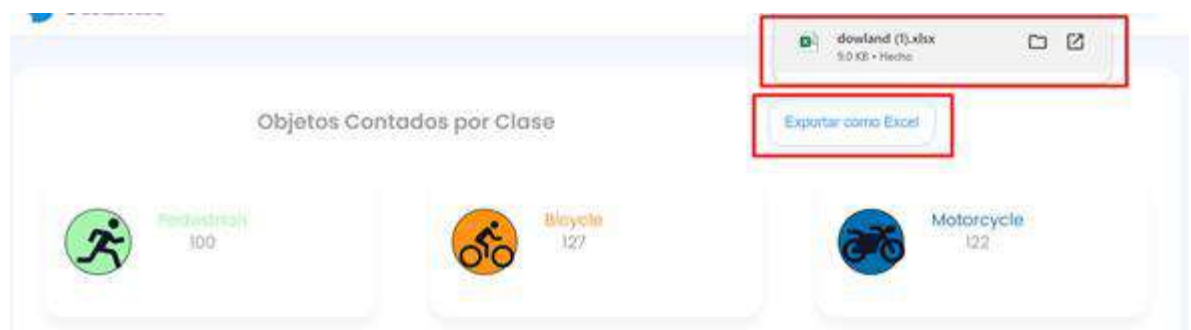
```
smart-micro-mb > frontend > src > pages > extraPages > Components > Home > JS StructureByClass.js > StructureBy
39     <div>{count}</div>
40   </div>
41 </div>
42 <div className="card_block1 cardh2">
43   <div className="card-icon "><IoIosBicycle /></div>
44   <div className="card-font">
45     <div className="font-description">{description}</div>
46     <div>{count}</div></div>
47 </div>
48 <div className="card_block1 cardh3">
49   <div className="card-icon"><FaMotorcycle /></div>
50   <div className="card-font">
51     <div className="font-description">{description}</div>
52     <div>{count}</div></div>
53 </div>
54 </div>
55 <div className="file-container">
56   <div className="card_block1 cardh4">
57     <div className="card-icon"><IoIosCar /></div>
58     <div className="card-font">
59       <div className="font-description">{description}</div>
60       <div>{count}</div></div>
61 </div>
62 <div className="card_block1 cardh5">
63   <div className="card-icon "><FaTruck /></div>
64   <div className="card-font">
65     <div className="font-description">{description}</div>
66     <div className="card-font">{count}</div>
67   </div>
68 </div>
69 <div className="card_block1 cardh6">
70   <div className="card-icon "><FaTruckMoving /></div>
71   <div className="card-font">
72     <div className="font-description">{description}</div>
73     <div className="card-font">{count}</div></div>
74 </div>
75 </div>
76 <div className="card_block1 cardh7">
77   <div className="card-icon"><FaQuestionCircle /></div>
```

```
smart-micro-mb > frontend > src > pages > extraPages > Components > Home > JS StructureByClass.js > StructureByClass
77     <div className="card-icon"><FaQuestionCircle /></div>
78     <div>
79       <div className="font-description">{description}</div>
80       <div className="card-font">{count}</div></div>
81     </div>
82   </div>
83
84   <div className="white-container">
85     <p className="fontTitle"><IntlMessages id='pages.extraPages.track.title' /> </p>
86     {<AreaChartHome />}
87   </div>
88
89   <div className="white-container">
90     <p className="fontTitle"><IntlMessages id='pages.extraPages.statistics.countedbystatistics' /> </p>
91     <div className="file-container">
92       <div>{<ChartRectanglesHome />}</div>
93       <div className="circular-progress-statistics">
94         {<CircularProgressHome />}
95       <div className="font-count-total-statistics">{<TotalStatistics />}</div>
96     </div>
97   </div>
98 </div>
99
100 <div className="file-container white-container">
101   <div className="container-circular-progress">
102     <div className="file-container cards1">
103       <p className="fontTitle"><IntlMessages id='pages.extraPages.direction1.title' /> </p>
104       <div className="card-icon-dir1"><AiOutlineArrowUp /></div>
105     </div>
106     <div className="circular-progress-home">
107       {<CircularProgressHome />}
108     </div>
109     <div className="card-font-d1">{<DataCountDirection />}</div>
110   </div>
111   <div className="container-circular-progress">
112     <div className="file-container cards2">
113       <p className="fontTitle"><IntlMessages id='pages.extraPages.direction2.title' /> </p>
114       <div className="card-icon-dir2"><AiOutlineArrowDown /></div>
115     </div>
116     <div className="circular-progress-home">
117       {<CircularProgressHome />}
118     </div>
119     <div className="card-font-d2">{<DataCountDirection />}</div>
120   </div>
121 </div>
122
123 <div className="white-container-center">
124   <p className="fontTitle"><IntlMessages id='pages.extraPages.statistics.linechart' /> </p>
125   {<DataLineChartbyStatistics />}
126 </div>
127 </>
128 );
129 }
130
131 export default StructureByClass;
```

4. Algunos de los estilos que se visualizan como hover, colores vibrantes, iconos para dar vida a la página son selección autónoma de experiencia de usuario del equipo de Scrum Team



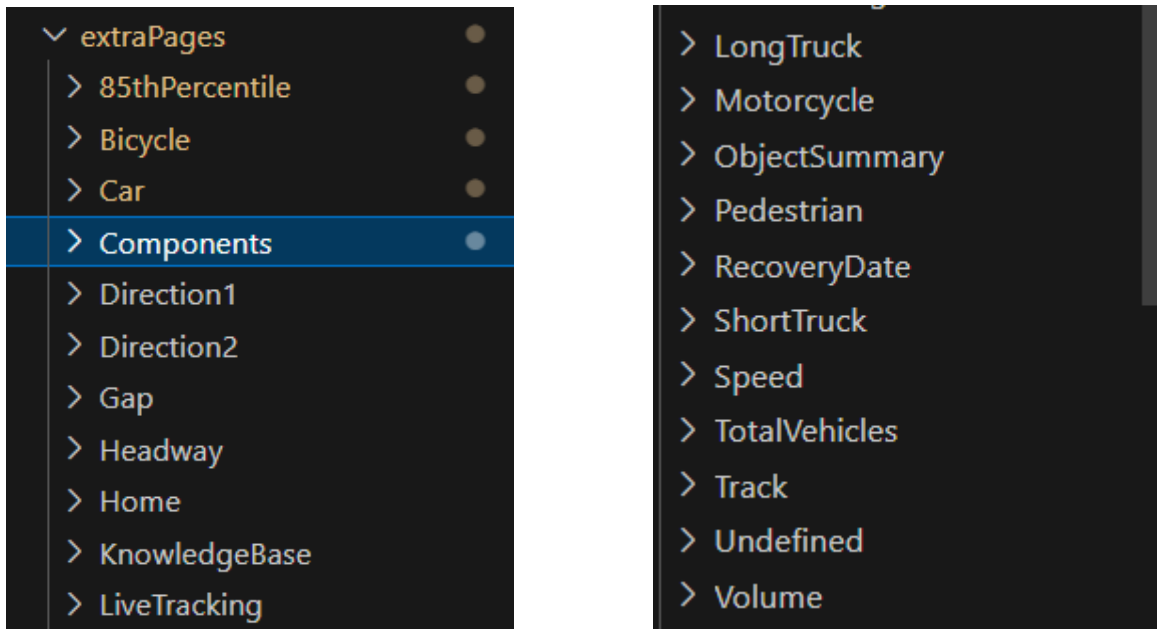
5. La funcionalidad de exportar datos a un archivo .xlsx y que estos datos fueran los mismos mostrados en pantalla, fue creado desde cero por el equipo de Scrum Team





En primer lugar, se debe tener en cuenta que las páginas directamente dentro de la carpeta extrapages ya estaban echas, por consiguiente, lo primero que se hizo fue crear la carpeta llamada componentes

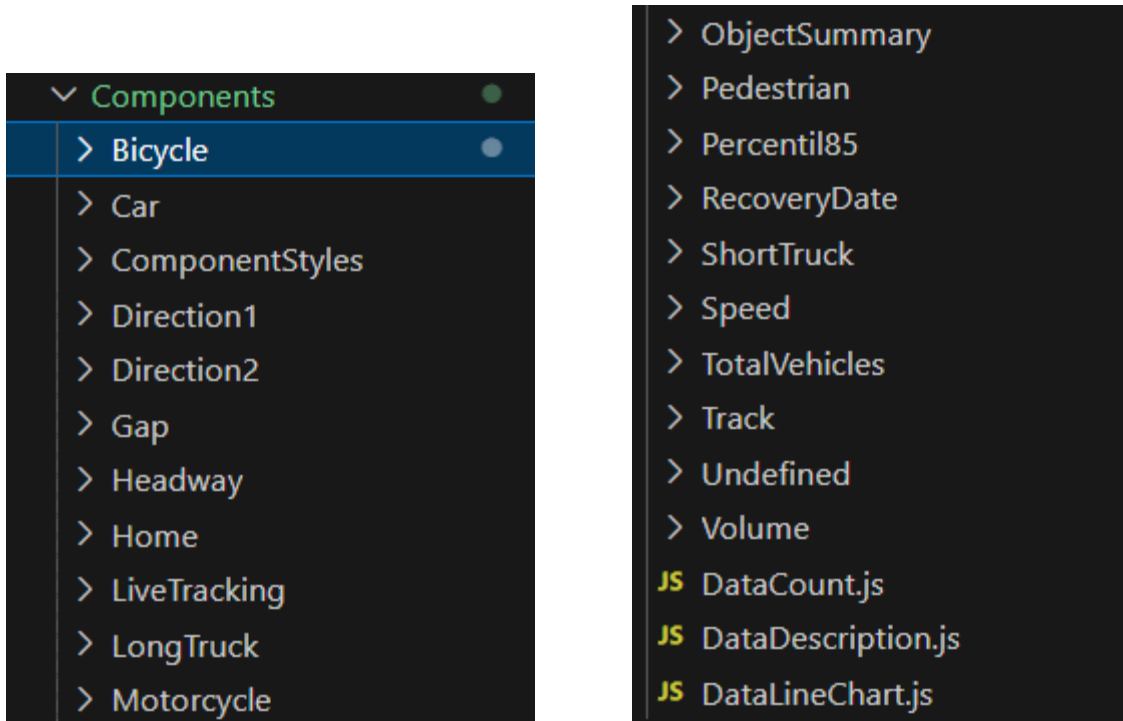
Figura 5. extraPages



Fuente: Autor

En segundo lugar, se creó cada una de las carpetas correspondientes a cada una de las páginas presentadas en el dashboard.

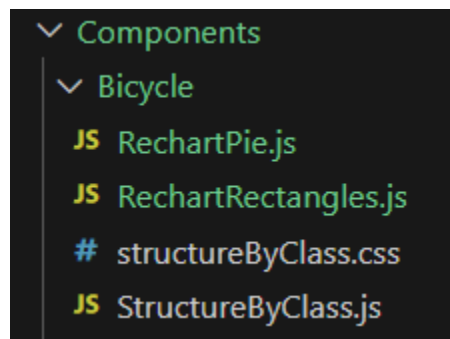
Figura 6. Páginas



Fuente: Autor

En tercer lugar, se puede apreciar que dentro cada carpeta de las páginas, se encuentran componentes individualizados de gráficas y estilos, que luego se utilizan dentro de la estructura final de la página en el componente StructureByClass.js. Como ejemplo a continuación, se presenta Bicicleta

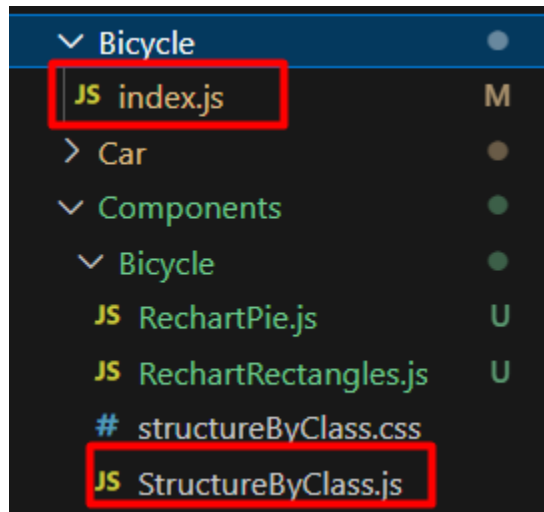
Figura 7. Página bicicleta



Fuente: Autor

En cuarto lugar, ya después de tener la estructura de la página final, se llama el componente de estructura final dentro del componente index, nombrado en un primer inicio, se sigue con el ejemplo de la bicicleta

Figura 8. Componente de estructura final e index



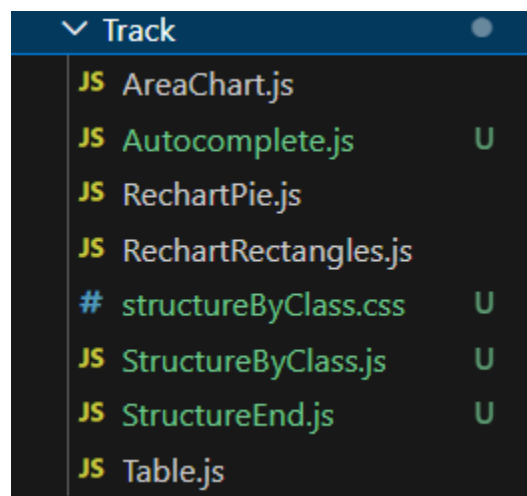
Fuente: Autor

Teniendo en cuenta la estructura del proyecto, se debe reiterar que el tercer y cuarto paso se repitió casi igual con cada una de las páginas, siguiendo el siguiente orden:

- En primer lugar, se hace el montaje de información gráfica correspondiente al módulo de tipos de vehículos, el cual está compuesto por las siguientes páginas: Peatón, bicicleta, motocicleta, carro, camión corto, camión largo e indefinido.
- En segundo lugar, se hace el montaje de información gráfica correspondiente al módulo de tipos de estadísticas, el cual está compuesto por las siguientes páginas: Volumen, velocidad, percentil 85, avance promedio y gap
- A continuación, se hace el montaje de la información gráfica de las siguientes páginas restantes: Home, Dirección, live tracking, total de vehículos, fecha de recuperación e histórico de objetos

- Seguidamente, se hace el montaje de la información gráfica del módulo carril. Este módulo es especial, puesto que en este se agregó un seleccionador el cual filtra los datos traídos desde la API, dependiendo del ítem seleccionado, estos ítems oscilan desde el carril1 al carri12. Por esta razón se puede notar con más componentes, que al final son utilizados en el componente de estructura.

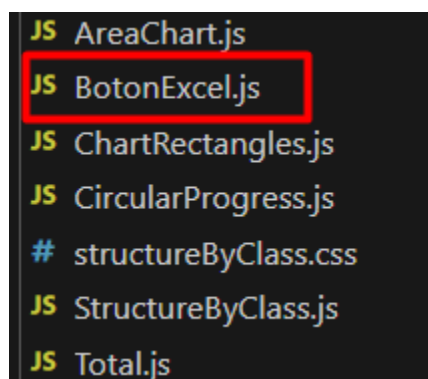
Figura 9. Página Carril



Fuente: Autor

- En quinto lugar, se hace el montaje del botón “Exportar como Excel”, el cual como su nombre lo indica, exporta los datos expuestos en el dashboard a Excel. Este componente se encuentra alojado en el home

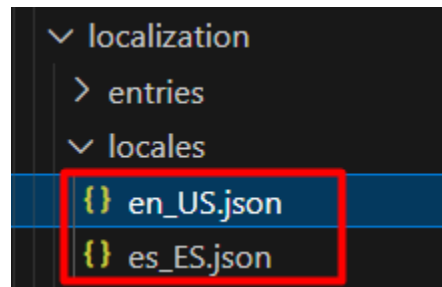
Figura 10. Botón Excel



Fuente: Autor

Por último, se hacen las traducciones de textos que no provienen directamente desde la API (títulos o descripciones e ítems del menú), puesto que los textos que provienen de la API se quedan con el idioma de origen.

Figura 11. Traducciones



Fuente: Autor

Adicionalmente, se recalca que el equipo de software obtiene la información de endpoint suministrados por el cliente, los cuales traen datos dinámicos que son reflejados en el dashboard. A continuación, se presenta como ejemplo los datos traídos de track. Para revisar los datos que trae cada uno de los endpoint, la opción es descargarlos y por ende mostrarlos en algún editor

Figura 22. Ejemplo de datos sección 1

```
C: > Users > ASUS_COREI7 > Downloads > {} countObjectsTrack (1).json > ...
1 [{"id":1,"name":"Track 1","track":99},
2 {"id":2,"name":"Track 2","track":122},
3 {"id":3,"name":"Track 3","track":77},
4 {"id":4,"name":"Track 4","track":186},
5 {"id":5,"name":"Track 5","track":145},
6 {"id":6,"name":"Track 6","track":67},
7 {"id":7,"name":"Track 7","track":81},
8 {"id":8,"name":"Track 8","track":137},
9 {"id":9,"name":"Track 9","track":61},
10 {"id":10,"name":"Track 10","track":63},
11 {"id":11,"name":"Track 11","track":139},
12 {"id":12,"name":"Track 12","track":60}]
```

Fuente: Autor

Figura 33. Ejemplo de datos sección 2

```
C: > Users > ASUS_COREI7 > Downloads > {} countObjectsTrack (3).json > ...
1 [{"id":1,"name":"Track 1","track":136},
2 {"id":2,"name":"Track 2","track":192},
3 {"id":3,"name":"Track 3","track":166},
4 {"id":4,"name":"Track 4","track":106},
5 {"id":5,"name":"Track 5","track":57},
6 {"id":6,"name":"Track 6","track":57},
7 {"id":7,"name":"Track 7","track":162},
8 {"id":8,"name":"Track 8","track":200},
9 {"id":9,"name":"Track 9","track":191},
10 {"id":10,"name":"Track 10","track":186},
11 {"id":11,"name":"Track 11","track":70},
12 {"id":12,"name":"Track 12","track":97}]
```

Fuente: Autor

## 6.5. Fase 5. Pruebas

Describir las pruebas aplicadas y el resultado.

Tabla 4. Pruebas

Tipo de prueba	Objetivo	Descripción	Resultados
Prueba de integración	Aprobar el enlace entre la API y la página de inicio	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de inicio, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Examinar el enlace entre la API y la página de dirección 1	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de dirección 1, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de dirección 2	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de dirección 2, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Live Tracking	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Live Tracking, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Carril	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Carril, se muestran los datos que corresponden a los Endpoint especificados en cada caso

Prueba de componentes	Comprobar el seleccionador de la página Carril	Evidenciar que los datos filtrados y mostrados por el seleccionador de la página Carril sean acordes al ítem escogido	Cada uno de los ítems que pertenecen al seleccionador de la página Carril, cumplen satisfactoriamente con mostrar únicamente los datos que se piden. Siendo estos ítems enumerados desde el Carril1 hasta el carril12
Prueba de integración	Aprobar el enlace entre la API y la página de Total de vehículos	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Total de vehículos, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de componentes	Comparar el resultado de la sumatoria de todos los vehículos, con el número de vehículos mostrados en las gráficas; en la página Total de vehículos	Comprobar que la sumatoria de todos los vehículos y el número de vehículos mostrados en las gráficas, concuerdan y llegan al mismo resultado	El resultado de la sumatoria del total de vehículos concuerda con el número de vehículos mostrados gráficamente
Prueba funcional	Comprobar el correcto funcionamiento del seleccionador incorporado en el ítem del menú Tipo de vehículos	Evidenciar que los datos filtrados y mostrados por el seleccionador incorporado en el ítem del menú Tipo de vehículos, sean acordes al ítem escogido	Cada uno de los ítems que pertenecen al seleccionador incorporado en el ítem del menú Tipo de vehículos, cumplen satisfactoriamente con mostrar únicamente los datos que se piden. Siendo estos ítems enumerados: Peatón, Bicicleta, Motocicleta, Carro, Camión corto, Camión largo, Indefinido
Prueba de integración	Aprobar el enlace entre la API y la página de Peatón	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Peatón, se muestran los datos que corresponden a

			los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Bicicleta	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Bicicleta, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Motocicleta	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Motocicleta, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Carro	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Carro, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Camión Corto	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Camión Corto, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Camión Largo	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Camión Largo, se muestran los datos que corresponden a los Endpoint especificados en cada caso
Prueba de integración	Aprobar el enlace entre la API y la página de Total de Indefinido	Verificar que se muestren correctamente los datos obtenidos a través de un Endpoint específico	En cada uno de los elementos de la página de Indefinido, se muestran los datos que corresponden a los Endpoint especificados en cada caso

Prueba de componentes	Comprobar el correcto funcionamiento del selector incorporado en el ítem del menú Tipo de estadísticas	Evidenciar que los datos filtrados y mostrados por el selector incorporado en el ítem del menú Tipo de estadísticas, sean acordes al ítem escogido	Cada uno de los ítems que pertenecen al selector incorporado en el ítem del menú Tipo de estadísticas, cumplen satisfactoriamente con mostrar únicamente los datos que se piden. Siendo estos ítems enumerados: Volumen, Velocidad, Percentil 85, Avance Promedio, Gap.
Prueba de componentes	Comprobar la fecha mostrada en la página Fecha de recuperación	Verificar que la fecha mostrada en la página Fecha de recuperación, corresponde a la fecha en la que está revisando el dashboard	La fecha mostrada en la página fecha de recuperación coincide con la fecha en la que se está revisando el dashboard
Prueba de componentes	Constatar la lista de fechas mostradas en la página Histórico de objetos	Comprobar que la lista de fechas presentadas en la Página Histórico de objetos, correspondan a las fechas en las que se accedió al dashboard recientemente	La lista de fechas que se muestran en la página Histórico de objetos, coincide con las fechas en las que se ingresó al dashboard recientemente
Prueba de componentes	Comprobar que el botón Exportar como Excel, funcione correctamente	Corroborar que el botón Exportar como Excel, funcione debidamente haciendo la descarga del archivo y exportando en el formato .xlsx	El botón Exportar como Excel cumple con la función de descargar el archivo en formato .xlsx
Prueba de componentes	Evidenciar que los datos descargados corresponden a los datos mostrados gráficamente	Verificar que los datos descargados en el archivo Excel, coincidan con los datos mostrados recientemente en el dashboard	Los datos descargados en el Excel concuerdan con los datos mostrados recientemente en el dashboard
Prueba de integración	Corroborar las traducciones de todas las páginas	Aprobar las traducciones hechas a textos que no vienen directamente de la API	Los textos como títulos y descripciones que no vienen directamente de la API, fueron traducidos satisfactoriamente

Fuente: Autor

## 7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

En el presente proyecto se desarrolló un dashboard en el cual se presentan gráficamente datos traídos de una API a través de varios endpoint. El proyecto fue exitosamente concluido puesto que, el desarrollo incluye principalmente ayudar al cliente a visibilizar dicha información en un dashboard ordenado y con estadísticas atractivas al usuario, este proceso se culminó satisfactoriamente. Finalmente, puedo concluir que el trabajar en este proyecto fue enriquecedor para mi persona, tanto en un ámbito de crecimiento académico como en un ámbito laboral de trabajo en equipo.

En el proceso de levantamiento de requerimientos, se procuró utilizar diferentes recursos para definir funcionalidades; por parte del equipo de desarrollo se explicó el alcance al que el proyecto puede llegar y por parte del cliente se expusieron las características imprescindibles para un buen funcionamiento del dashboard. Del mismo modo, para el tema de mockups se concertó con el cliente la facilidad de no hacerlos, y en su lugar tomar como guía la plantilla crema para aprovechar el tiempo de desarrollo. El anterior proceso se hizo siempre de la mano con el cliente, el cual quedó satisfecho con la lista resultante de funcionalidades y características; seguidamente se hace implementación del componente FrontEnd, se focalizó en seguir y cumplir la lista de requisitos previamente concertada con el cliente, terminando a cabalidad todas las funcionalidades; De lo anterior se obtiene un dashboard ordenado y funcional que visualiza estadísticas y hace conteo de vehículos, por lo que se concluye exitosa esta fase.

El proceso de validación se ejecutó mediante 2 tipos de pruebas. Las primeras fueron las de integración que se aplicaron para comprobar el enlace entre los datos de la API y cada una de las páginas, con lo cual se corroboró que efectivamente hubo flujo de datos y además que los datos mostrados coinciden con lo que se estaba pidiendo. Las siguientes pruebas que se hicieron fueron las funcionales, que primeramente se aplicaron con el elemento seleccionador, probando que los ítems si son los indicados y filtran correctamente la información; seguidamente con el elemento botón exportar, verificando que si se hace proceso de descarga y los datos coinciden con los que se muestran gráficamente; por último, se hizo esta prueba con la fecha de recuperación y el histórico de objetos, verificando que efectivamente las fechas coinciden con los últimos accesos al dashboard.

## 8. REFERENCIAS

Axios. (2023). Axios Empezando. Axios. Recuperado el 14 de Diciembre de 2023 de <https://axios-http.com/es/docs/intro>

Bell, A. (mayo de 2021). Aprender CSS. Webdev. <https://web.dev/learn/css?hl=es>

Crema CRA. (noviembre de 2023). About Crema CRA. Crema React. Recuperado el 14 de Diciembre de 2023 de <https://docs.cremawork.com/>

Docker. (2023). Try Docker Compose. Docker Docs. Recuperado el 14 de Diciembre de 2023 de <https://docs.docker.com/compose/gettingstarted/>

Garzías, J. [Javier Garzías] (26 enero de 2021). ¿Cuándo usar metodologías ágiles? vs Ágil vs Tradicional vs (más que metodologías... frameworks). YouTube. <https://www.youtube.com/watch?v=ZACy8q2JdVM>

GCFAprendeLibre [GCFAprendeLibre] (2021). Crea tu equipo de trabajo con Scrum I Curso de Scrum. YouTube. <https://www.youtube.com/watch?v=v9yhSINa1KY&t=3s>

González, A. (Marzo de 2021). Evolución Histórica de los Dashboards. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Laoyan, S. (26 de agosto de 2022). ¿Qué es un Project manager y cómo convertirte en uno? Asana. <https://asana.com/es/resources/become-a-project-manager>

Martínez, Juan. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. RA-MA.

Navarro, A., Fernández, J., Morales, J. (Diciembre de 2013). Revisión de metodologías ágiles para el desarrollo de software. <https://www.redalyc.org/pdf/4962/496250736004.pdf>

Node Js. (2023). About Node.js. Node Js. Recuperado el 14 de Diciembre de 2023 de <https://nodejs.org/en/about>

Pacienza, J., Maida, E. (Diciembre de 2015). Metodologías de desarrollo de software. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Parra, D., Vélez, J., Vázquez, S., Fandiño, A. (2018). Definición e implementación de dashboard para el seguimiento y control de indicadores en oficinas de nueva eps. <https://alejandria.poligran.edu.co/bitstream/handle/10823/1735/DEFINICI%C3%93N%20E%20IMPLEMENTACI%C3%93N%20DE%20DASHBOARD%20PARA%20EL%20SEGUIMIENTO%20Y%20CONTROL%20DE%20INDICADORES%20EN%20OFICINAS%20DE%20NUEVA%20EPS.pdf?sequence=1&isAllowed=y>

React Js. (2023). Quick Start. React Js. Recuperado el 14 de Diciembre de 2023 de <https://react.dev/learn>

Redux. (2023). Getting Started with Redux. Redux. Recuperado el 14 de Diciembre de 2023 de <https://es.redux.js.org/>

Schwaber, K., Sutherland, J. (Julio de 2013). La Guía de Scrum. <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Serna, Edgar. (2013). *Prueba funcional de software, Un proceso de verificación constante*. Fondo Editorial ITM.

Sommerville, Ian. (2006). *Ingeniería del software*. Pearson Addison Wesley.

Team Asana. (13 de abril de 2023). Cómo redactar un documento de requisitos de software Asana. <https://asana.com/es/resources/software-requirement-document-template>

Tristán, E. (27 de octubre de 2014). ¿Qué es un dashboard y para qué sirve?. Tristán Elósegui. <https://tristanelosegui.com/2014/10/27/que-es-y-para-que-sirve-un-dashboard/>

Visual Studio Code. (2023). Getting Started. Visual Studio Code Docs. Recuperado el 14 de Diciembre de 2023 de <https://code.visualstudio.com/docs>

Zapater, S. (13 de marzo de 2022). ¿Qué es un Scrum Master y cuál es su papel? Hiberusbolg. [https://www.hiberus.com/crecemos-contigo/que-es-scrum-master/#:~:text=Qu%C3%A9%20hace%20un%20Scrum%20Master%20y%20funciones&text=Es%20el%20encargado%20de%20identificar,entorno%20de%20trabajo%2C%20etc\).](https://www.hiberus.com/crecemos-contigo/que-es-scrum-master/#:~:text=Qu%C3%A9%20hace%20un%20Scrum%20Master%20y%20funciones&text=Es%20el%20encargado%20de%20identificar,entorno%20de%20trabajo%2C%20etc).)

## 9. ANEXOS

### Anexo 1. Ficha Técnica

Utilice esta sección para indicar las características técnicas del software, describa las condiciones que requiere un equipo de cómputo para su despliegue.

#### HERRAMIENTAS QUE SE DEBEN INSTALAR:

- Docker compose
- Node js
- Git
- SSD Unidad de estado sólido. Es recomendable para un buen funcionamiento del proyecto
- No existen restricciones respecto al navegador

## **Anexo 2. Manual de Usuario**

### **INTRODUCCIÓN**

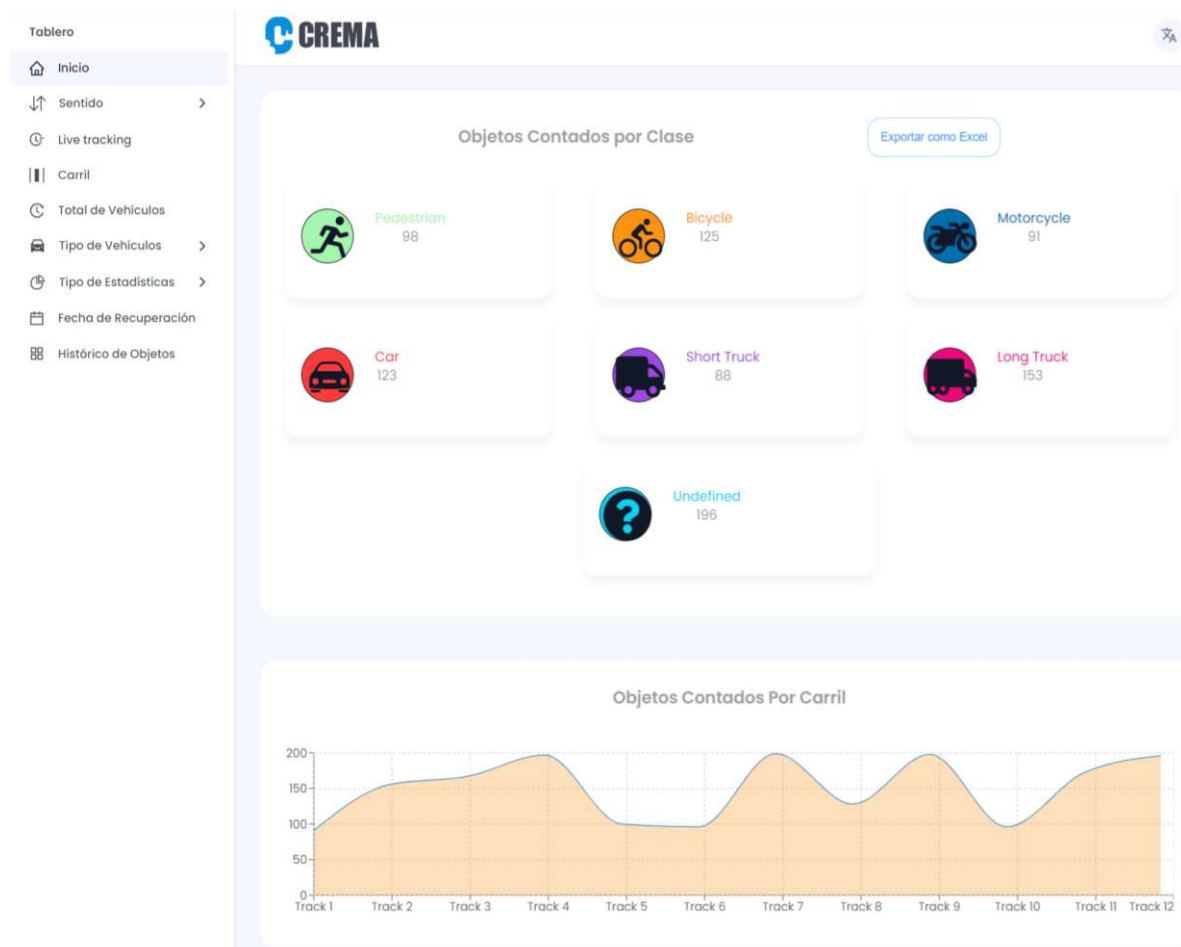
Este manual se encarga de guiar al usuario en su recorrido por el dashboard, facilitando la orientación y mejorando su experiencia. El dashboard permite la visualización gráfica de los datos obtenidos del radar de carretera, a través de diferentes páginas que en él se encuentran.

El objetivo de este manual es mostrar al usuario todas las características y funciones disponibles dentro del dashboard, mostrar paso a paso cada una de las páginas y así se pueda aprovechar correctamente esta herramienta.

## INICIO

Ingresar al dashboard es muy fácil, puesto que no existe login y por lo tanto no es necesario tener un usuario o contraseña para acceder. En la página de inicio se encuentra, primeramente, la sección de tipos de vehículos, donde se visualizan las estadísticas correspondientes.

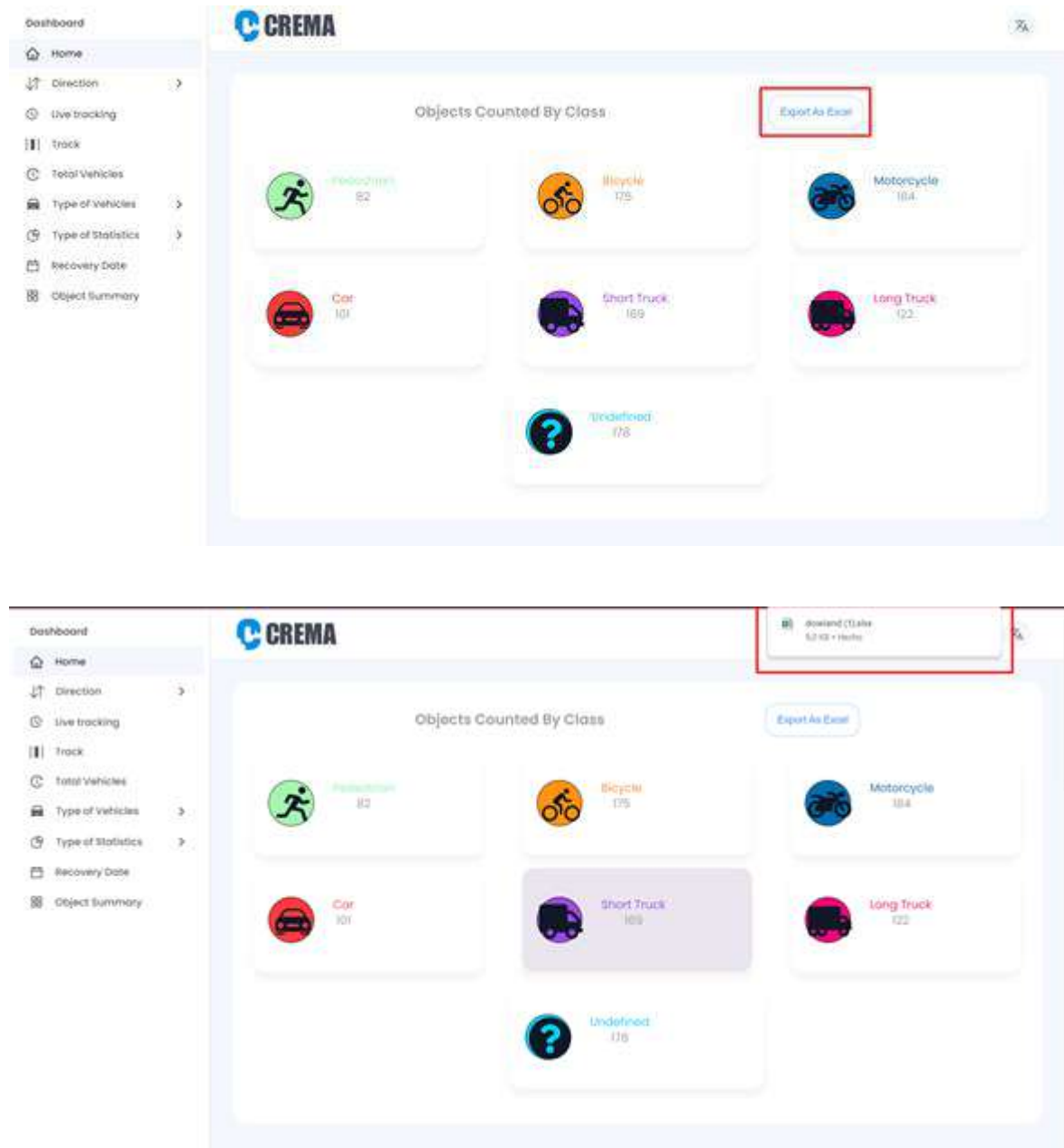
Figura 44. Inicio sección1



Fuente: Autor

Así mismo, se encuentra el botón “Exportar como Excel” que se encuentra en la parte superior derecha, con el cual se puede descargar los datos presentados visualmente como un archivo .xlsx.

Figura 55. Exportar como Excel



Fuente: Autor

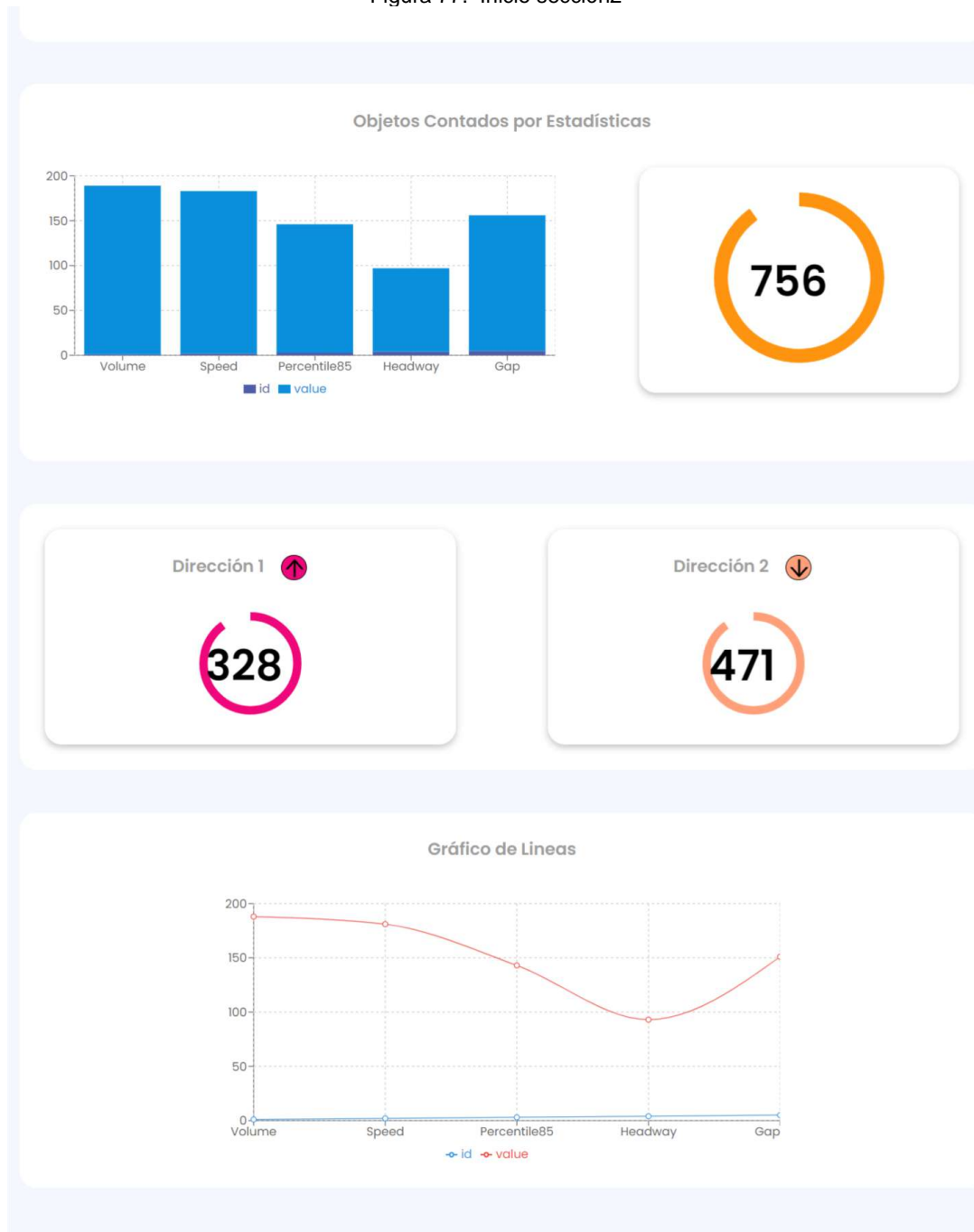
Figura 66. Archivo Excel

id	name	count
1	Pedestrian	82
2	Bicycle	175
3	Motorcycle	184
4	Car	101
5	Short Truck	169
6	Long Truck	122
7	Unidentified	178

Fuente: Autor

Seguidamente se encuentran la gráfica de área correspondiente a carriles, el gráfico rectangular y círculo de progreso correspondiente a datos de tipo de estadísticas, los datos correspondientes a dirección1 y dirección2, y por último se encuentra el gráfico de líneas correspondiente a tipo de estadísticas.

Figura 77. Inicio sección2



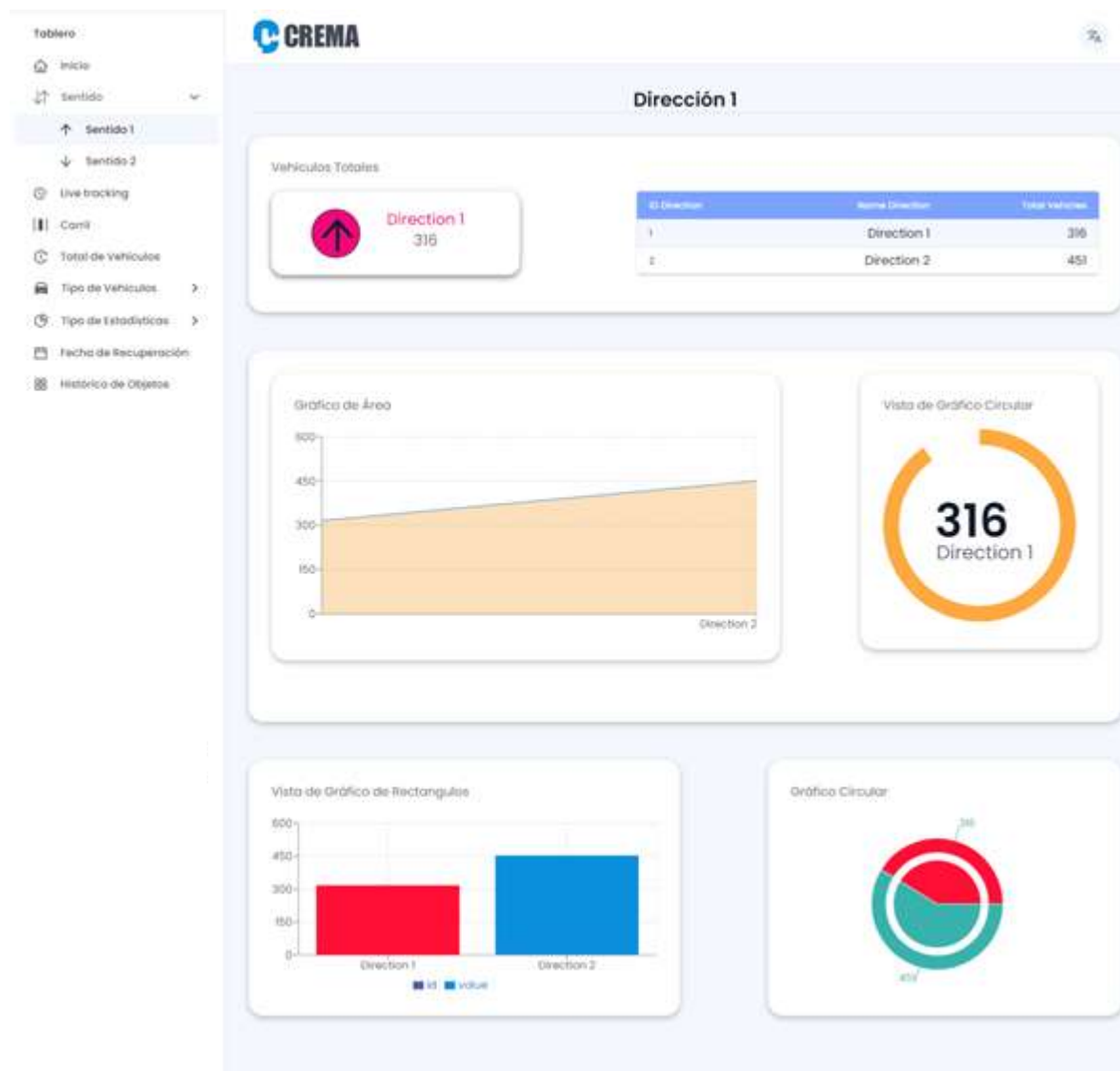
Fuente: Autor

## SENTIDO

El ítem “Sentido” ubicado en el menú lateral consta de 2 submenús: Sentido 1 y Sentido 2. Al ingresar a cada una de esas 2 páginas, se encuentra en la parte superior datos de los vehículos totales que han pasado por cada sentido correspondientemente. Seguidamente se encuentra un gráfico de área, un gráfico circular y una vista rectangular que muestran los datos anteriormente mencionados.

### SENTIDO1:

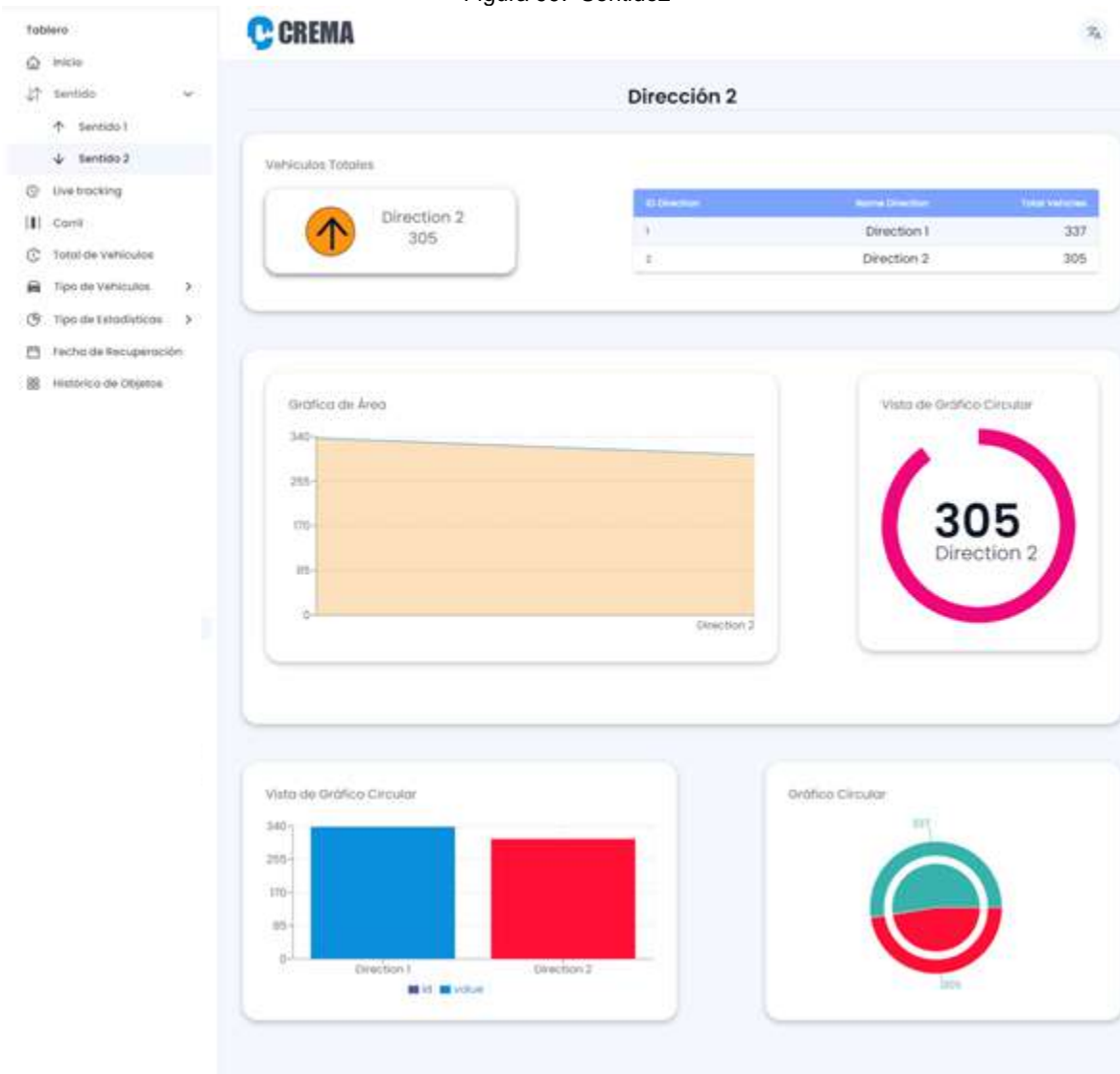
Figura 88. Sentido1



Fuente: Autor

SENTIDO2:

Figura 99. Sentido2



Fuente: Autor

## LIVE TRACKING

En esta página de “Live Tracking” se muestran datos obtenidos de todos los tipos de vehículos, mostrándolos en una vista agradable al usuario. Primeramente, se muestra un gráfico de líneas, seguido de un círculo de progreso con el total de vehículos, al lado se muestra un gráfico de área y por último se muestra un gráfico de rectángulos.

Figura 20. Live Tracking



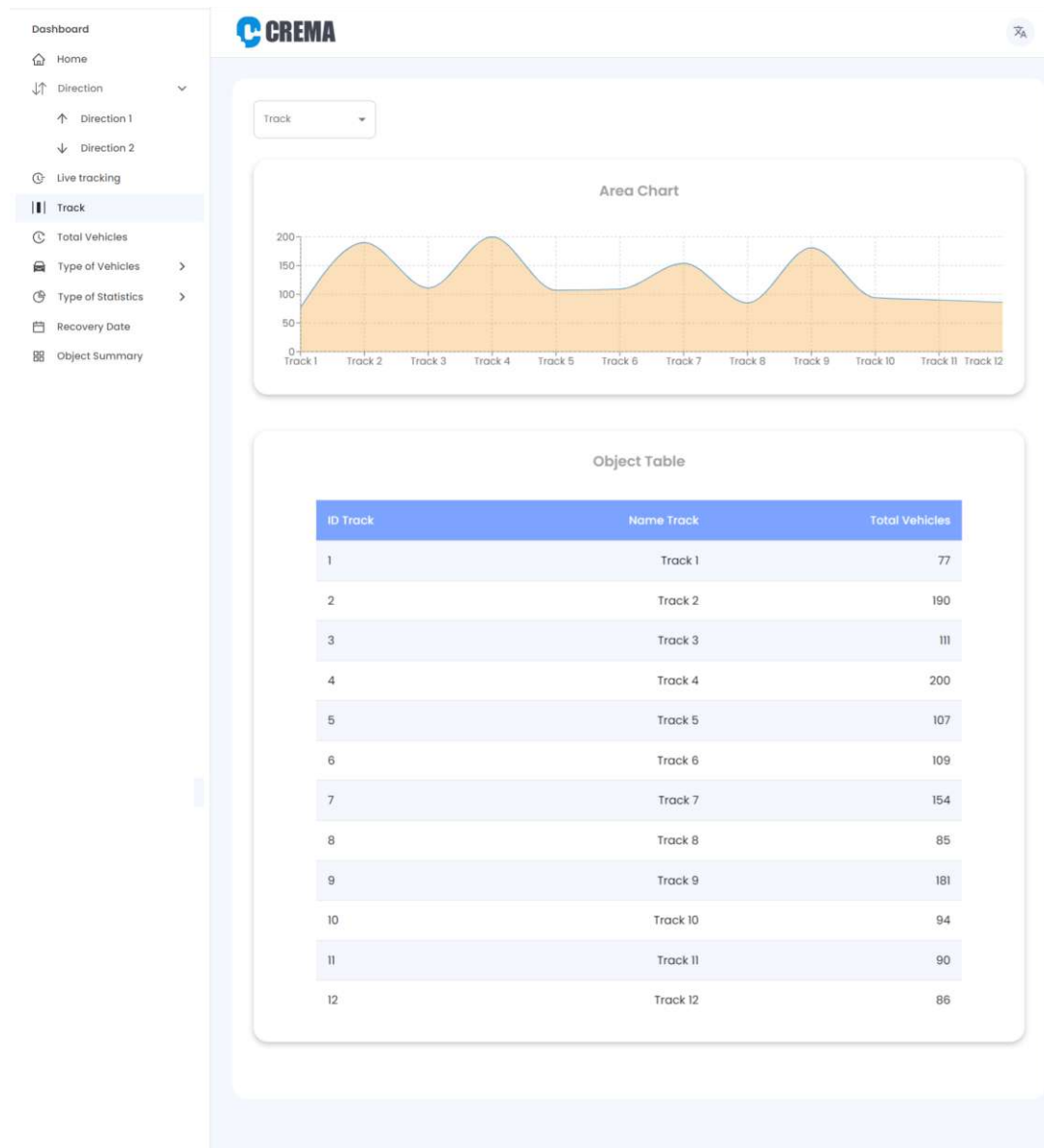
Fuente: Autor

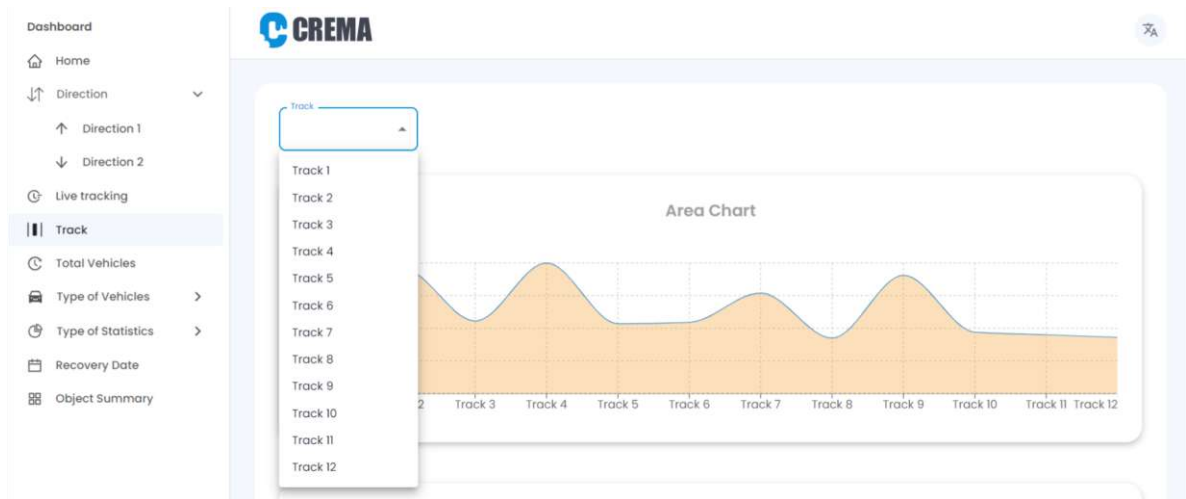
## CARRIL

En la página de “Carril” se muestran los datos correspondientes a cada uno de los carriles. Esta página tiene un seleccionador el cual está situado en la parte superior izquierda, funciona para filtrar la información de los ítems que se elijan respectivamente, siendo los ítems que este contiene numerados del Carril1 al Carril12.

### SELECCIONADOR CARRIL:

Figura 210. Seleccionador carril

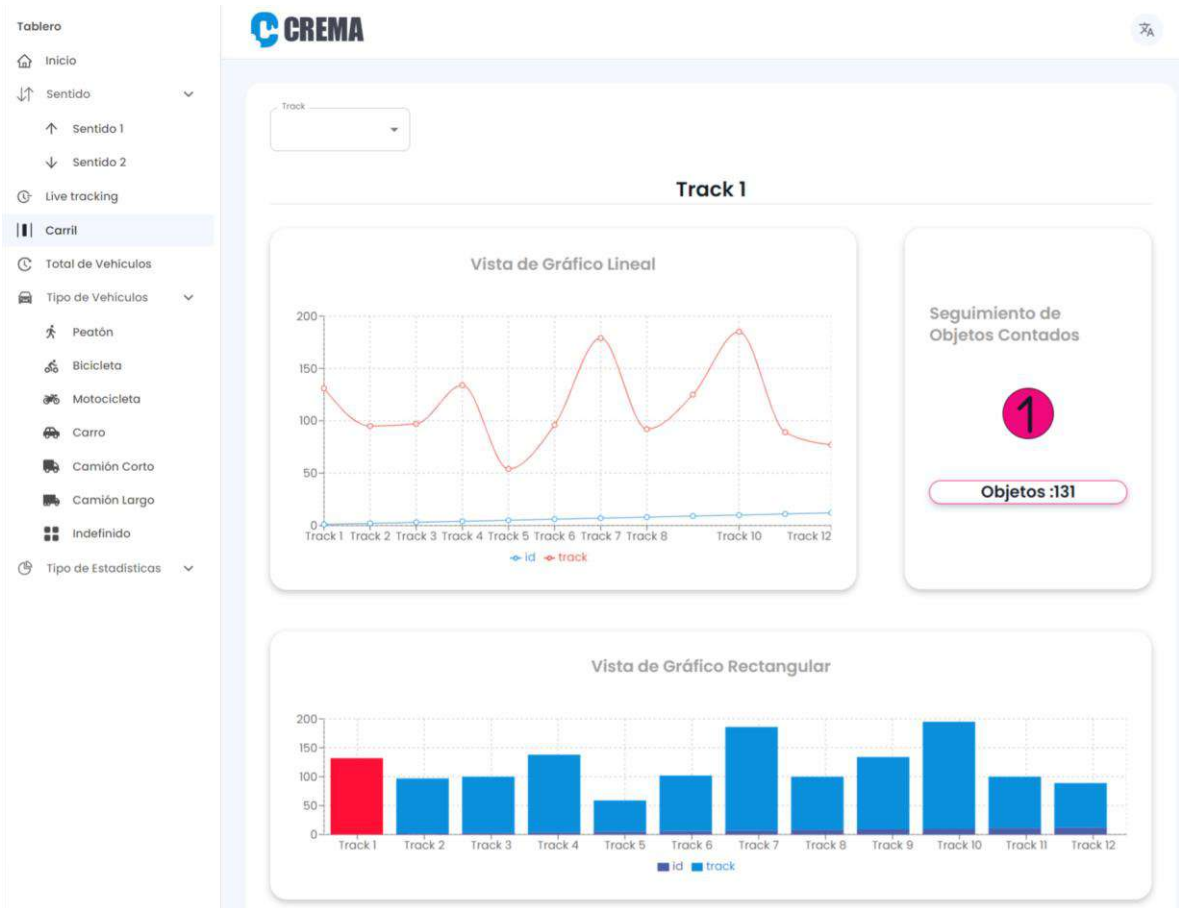




Fuente: Autor

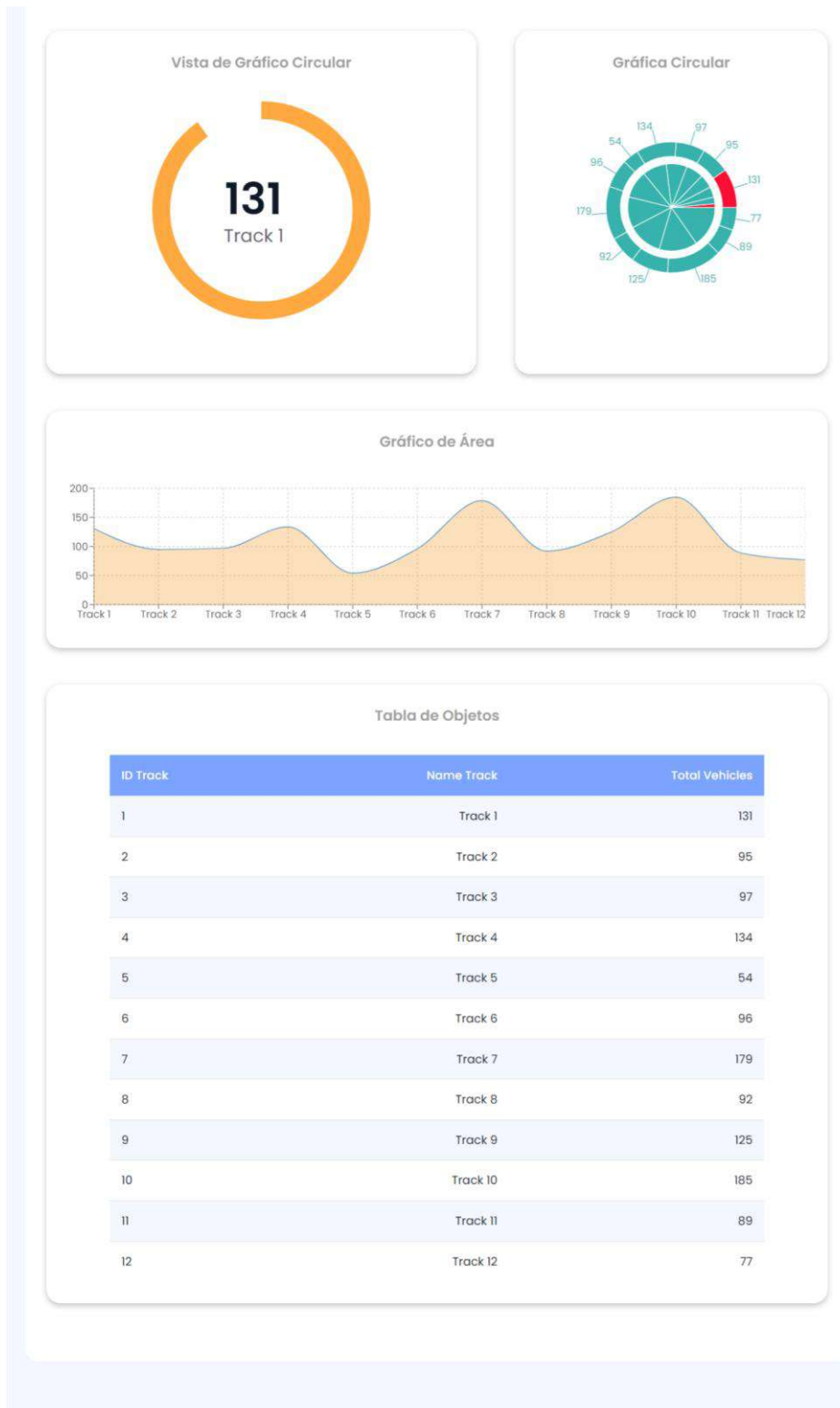
CARRIL1:

Figura 22. Carril1 sección1



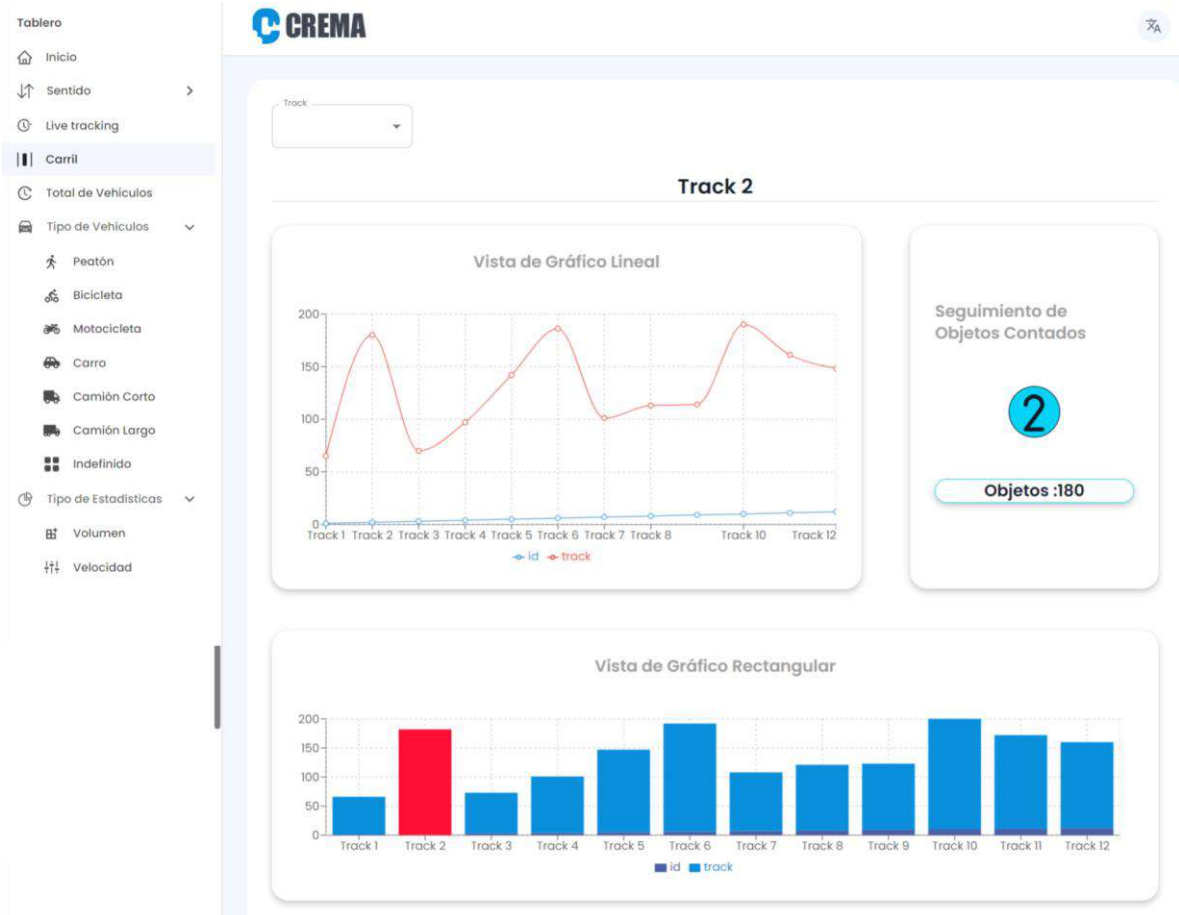
Fuente: Autor

Figura 23. Carril1 sección2



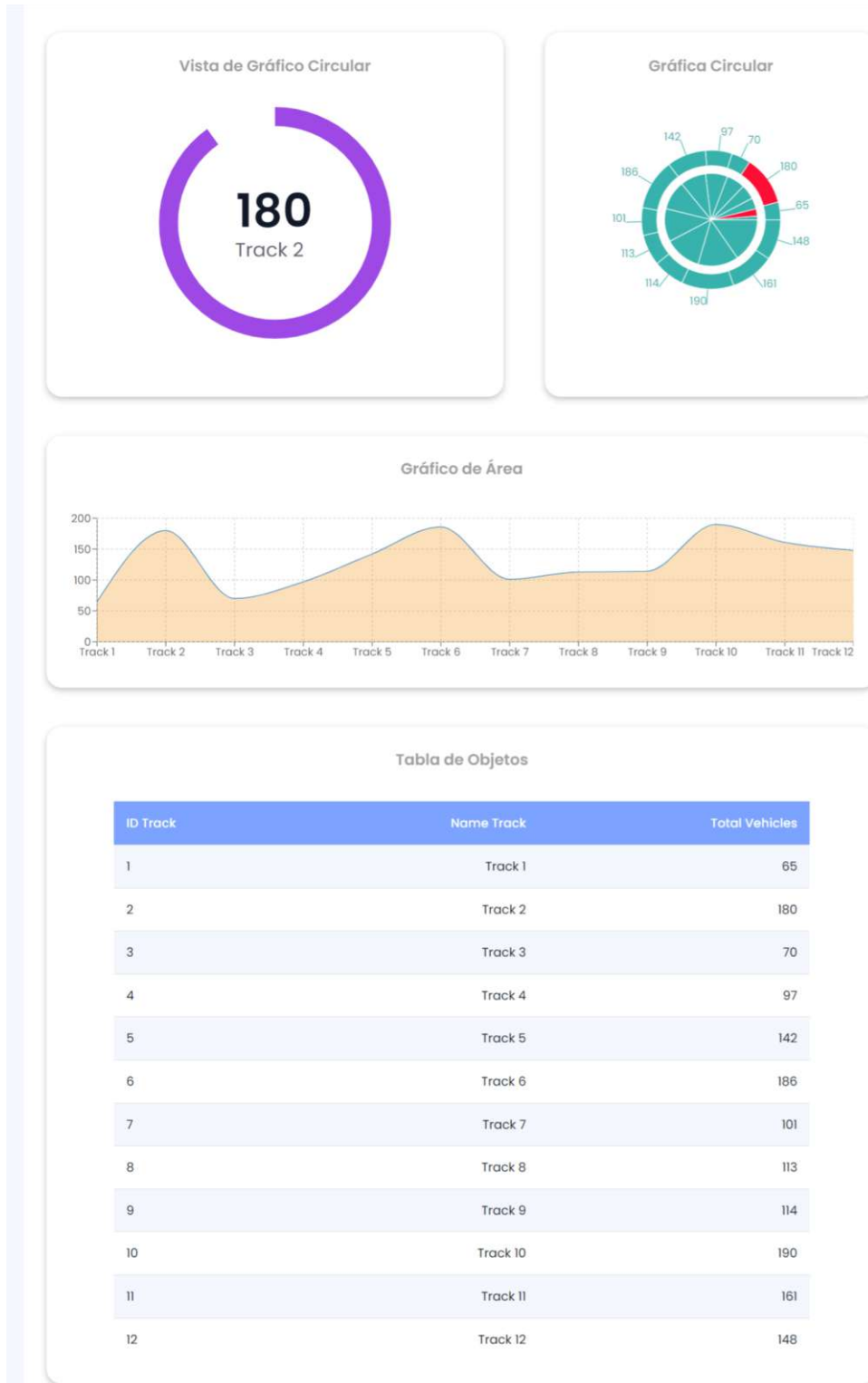
CARRIL2:

Figura 24. Carril2 sección1



Fuente: Autor

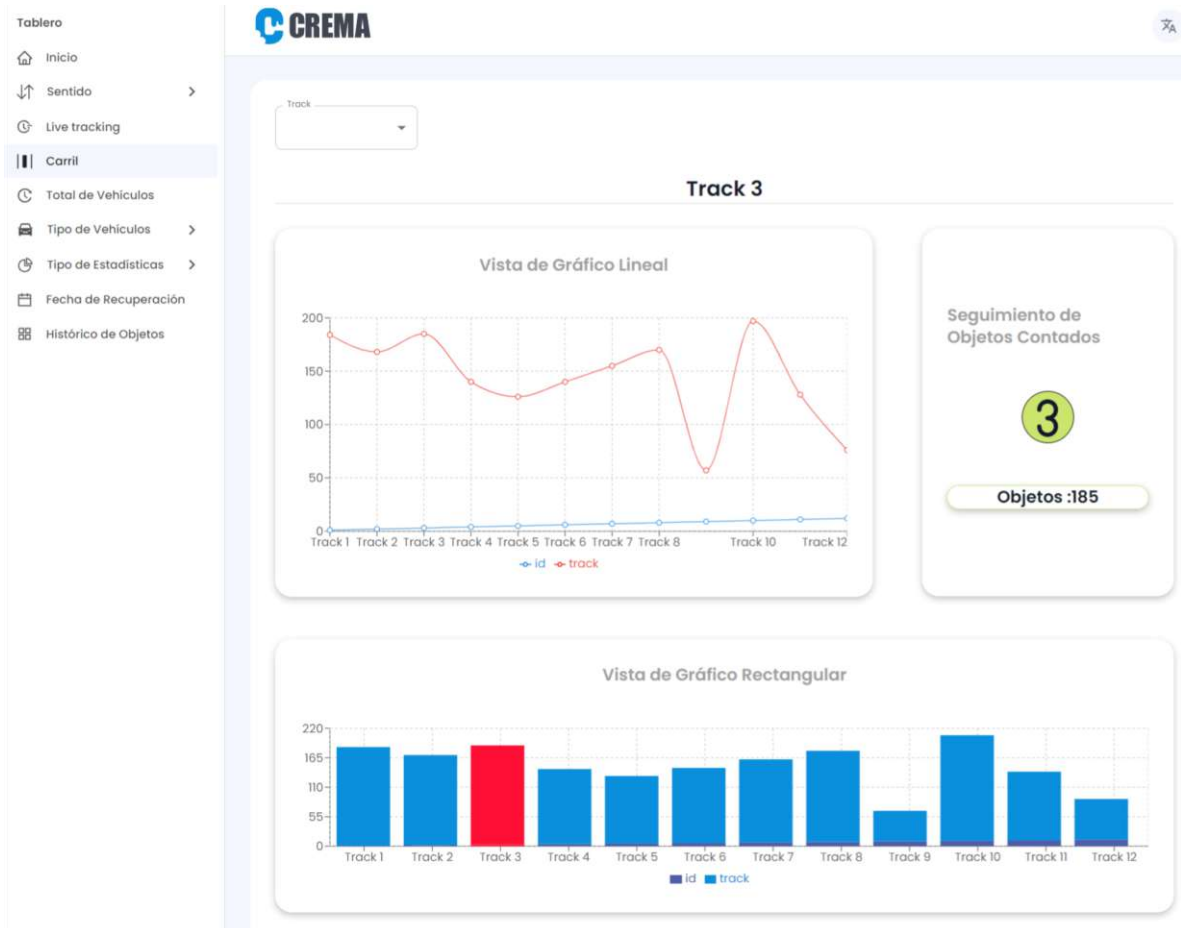
Figura 25. Carril2 sección2



Fuente: Autor

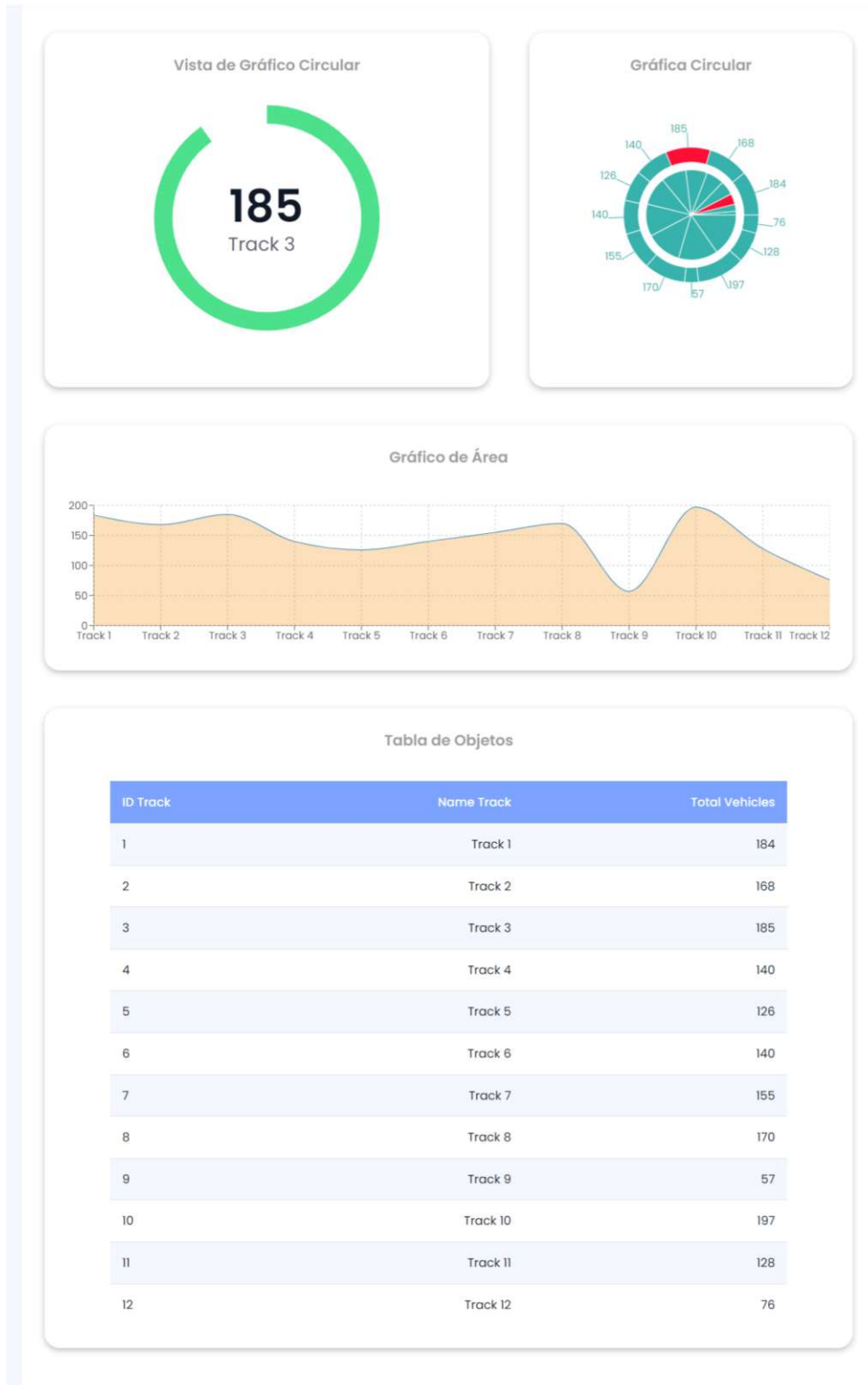
CARRIL3:

Figura 26. Carril3 sección1



Fuente: Autor

Figura 27. Carril3 sección2



Fuente: Autor

CARRIL4:

Figura 28. Carril4 sección1

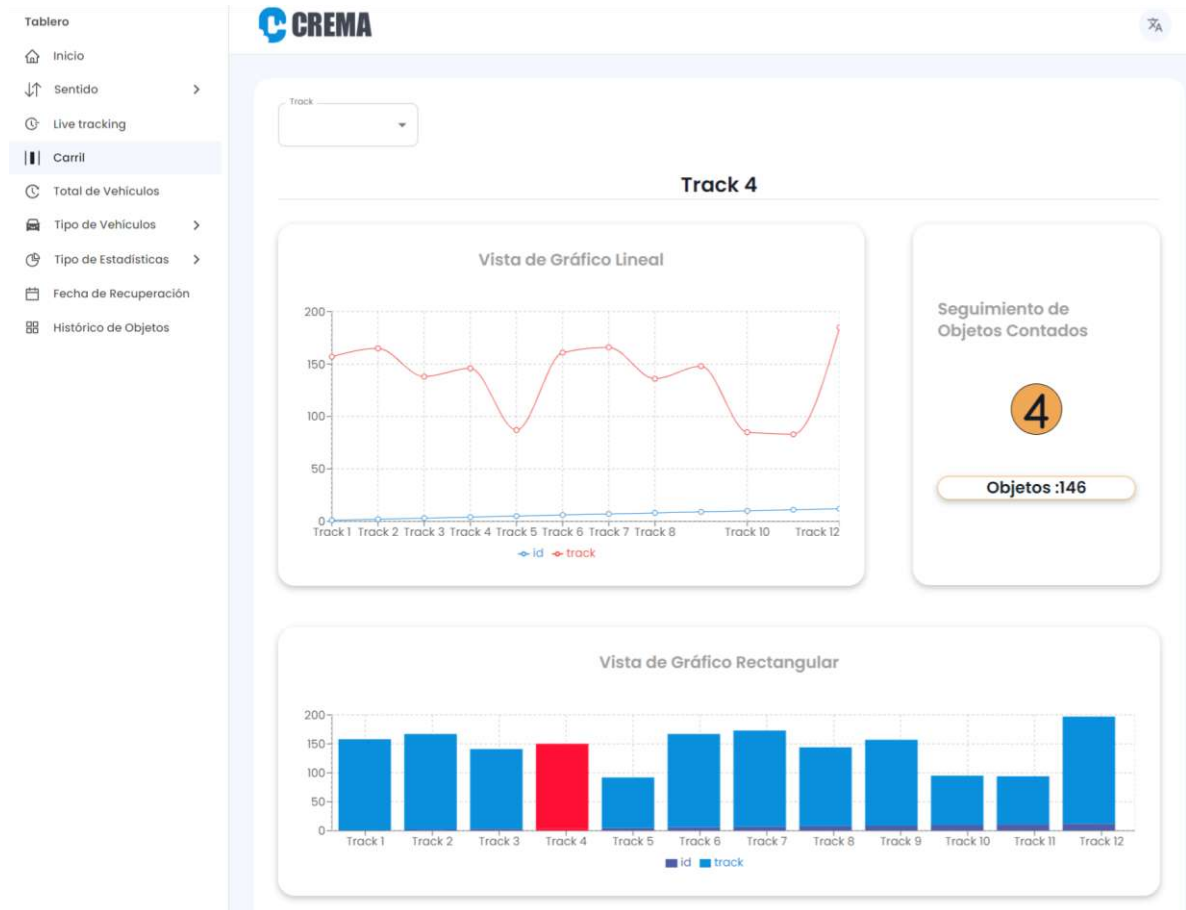
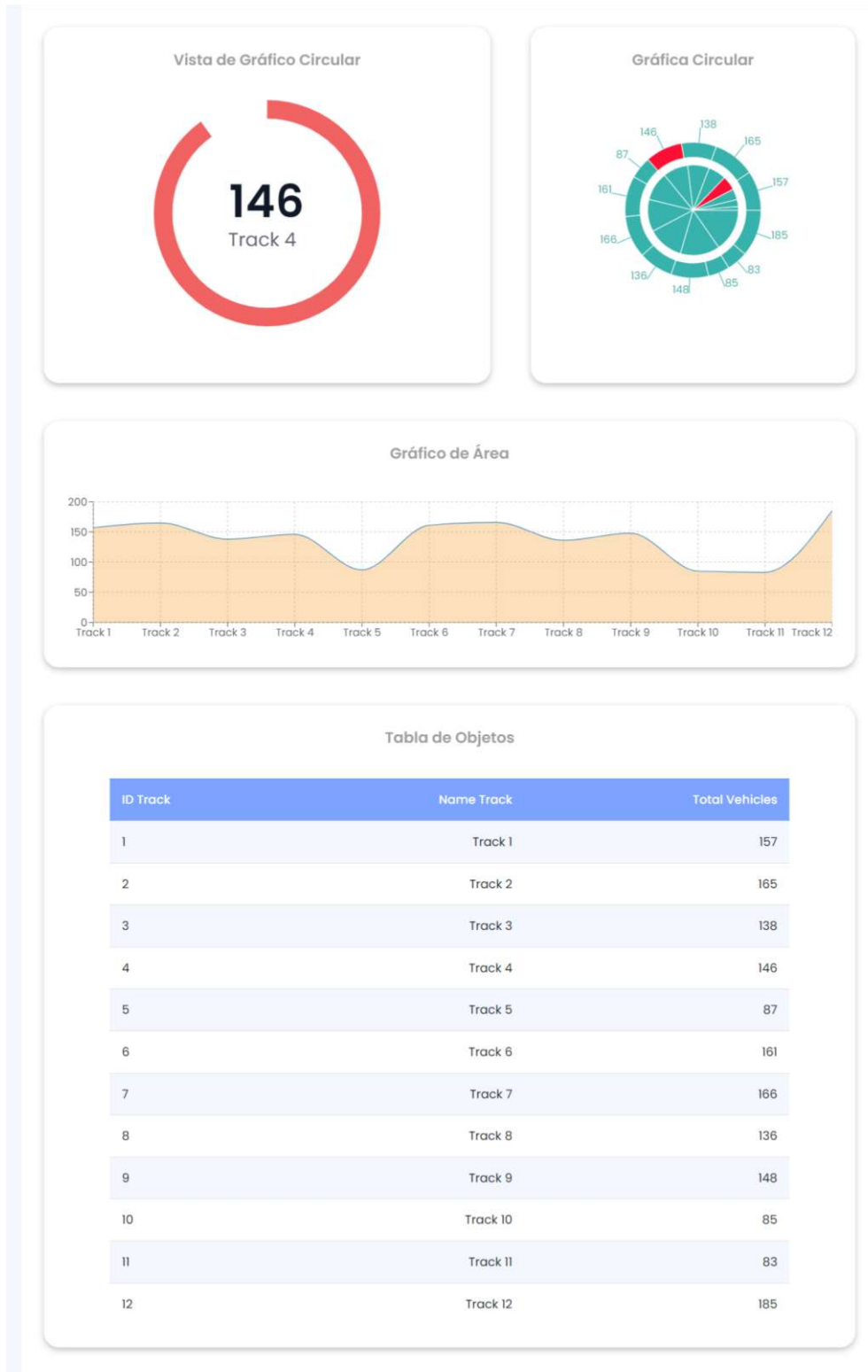


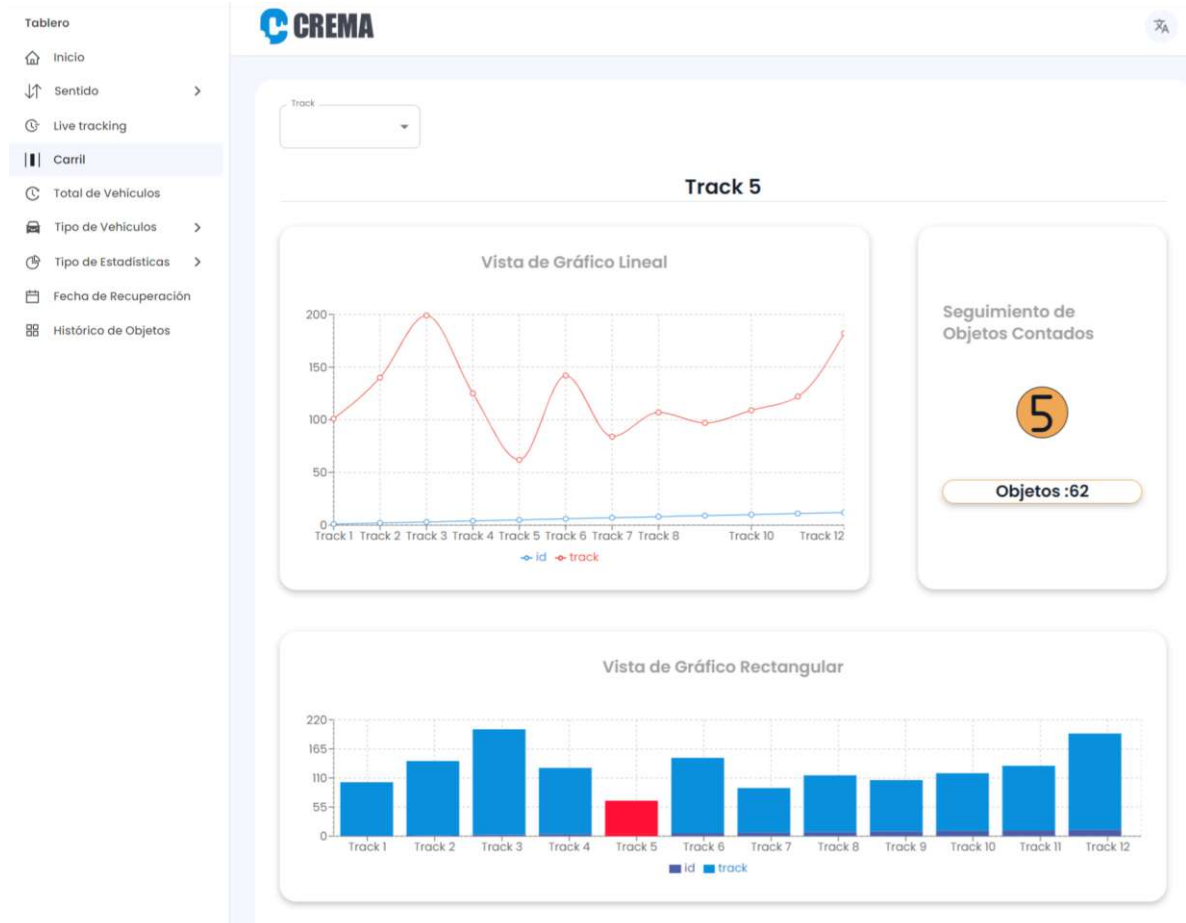
Figura 29. Carril4 sección2



Fuente: Autor

CARRIL5:

Figura 30. Carril5 sección1



Fuente: Autor

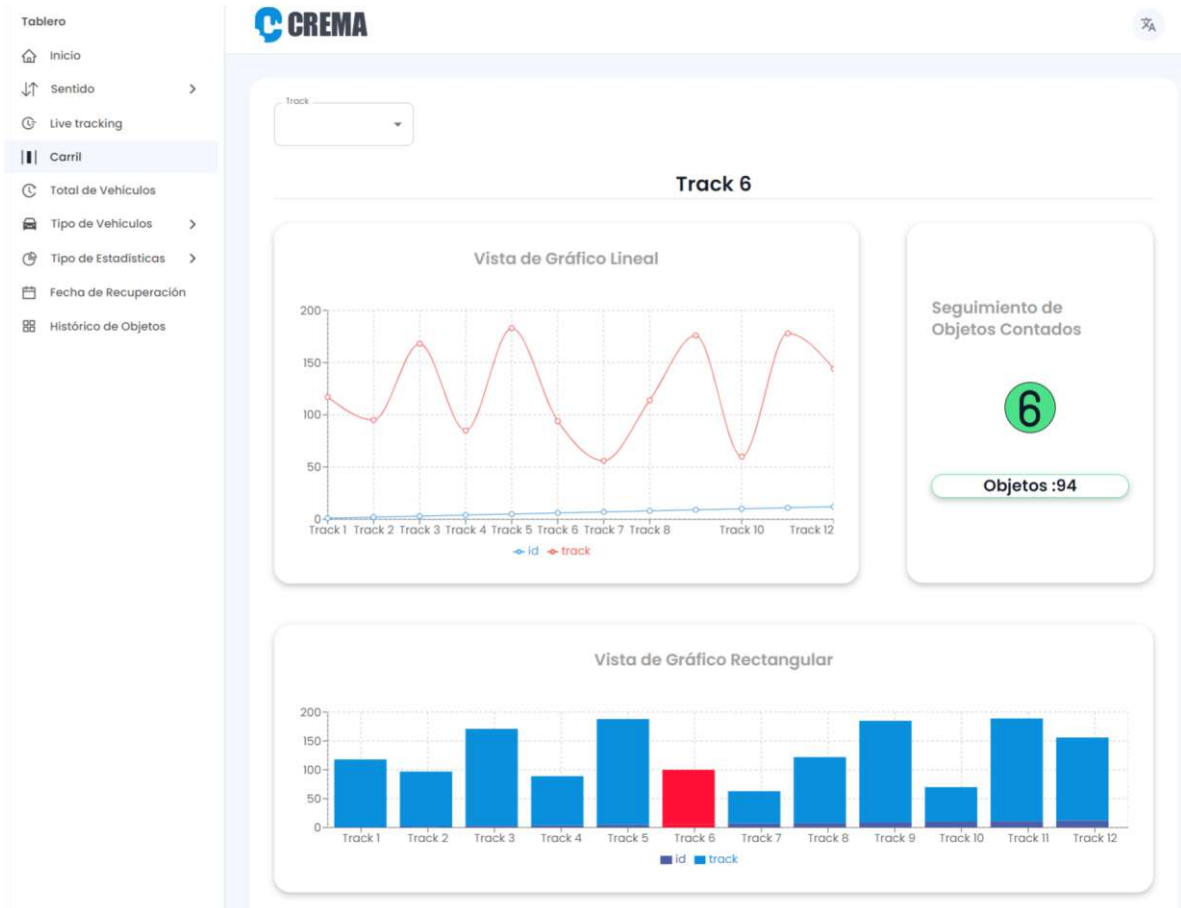
Figura 31. Carril5 sección2



Fuente: Autor

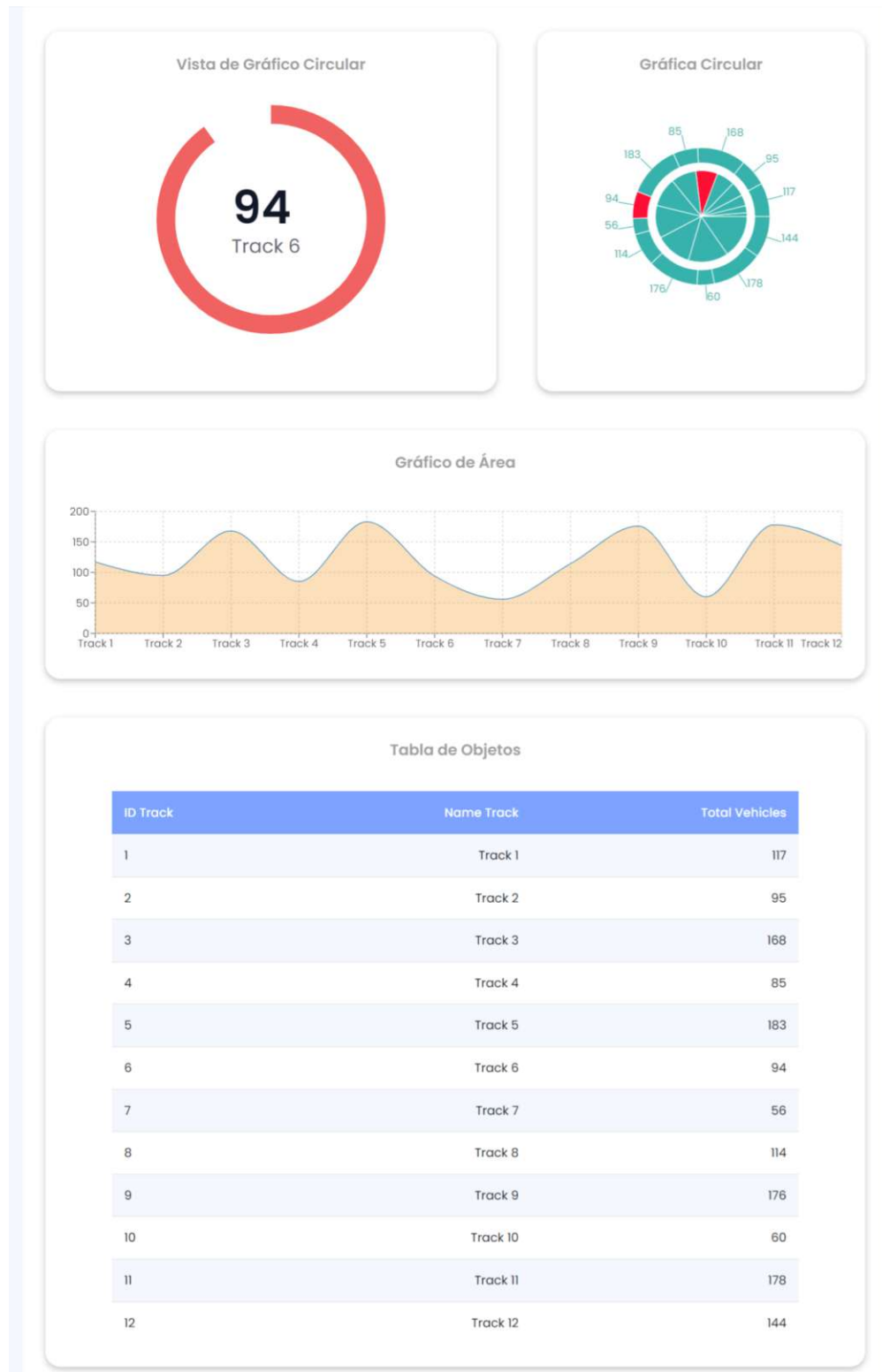
CARRIL6:

Figura 32. Carril6 sección1



Fuente: Autor

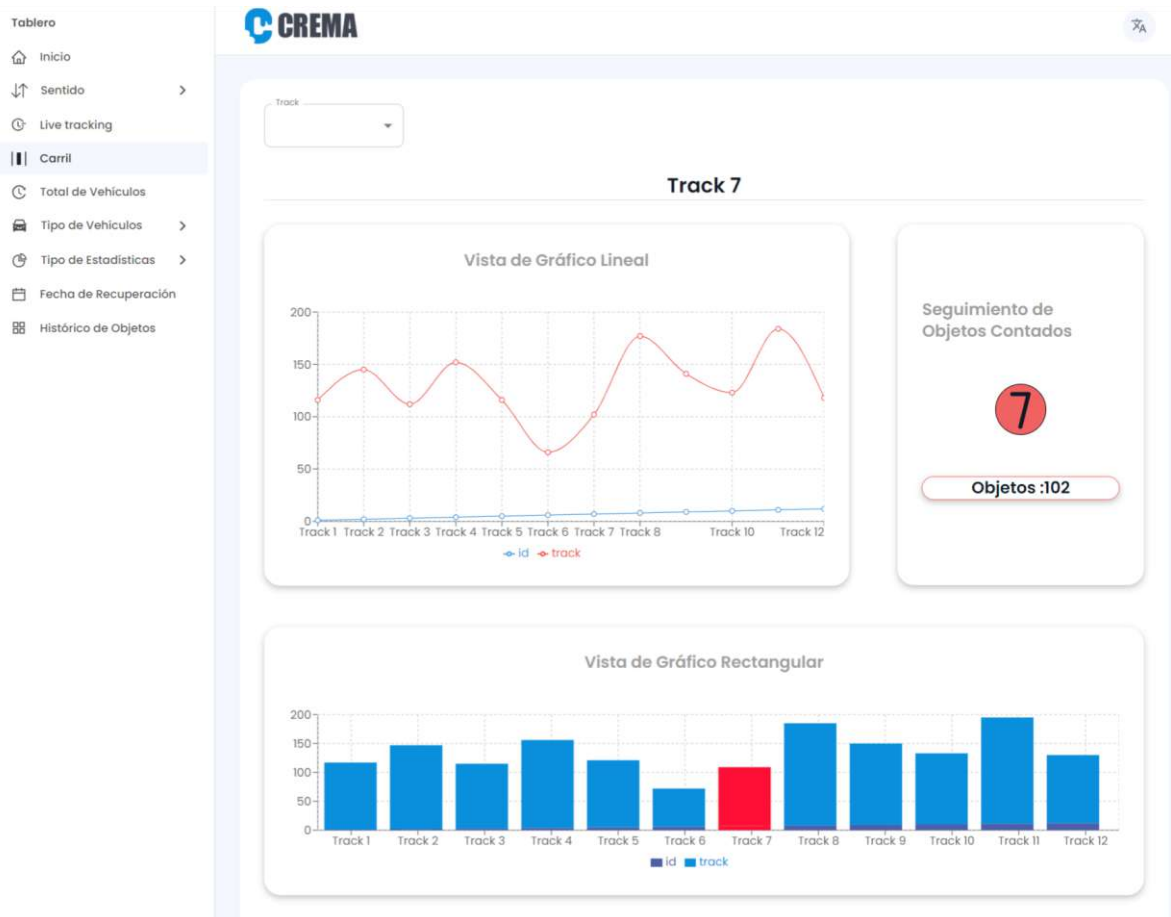
Figura 33. Carril6 sección2



Fuente: Autor

CARRIL7:

Figura 34. Carril7 sección1



Fuente: Autor

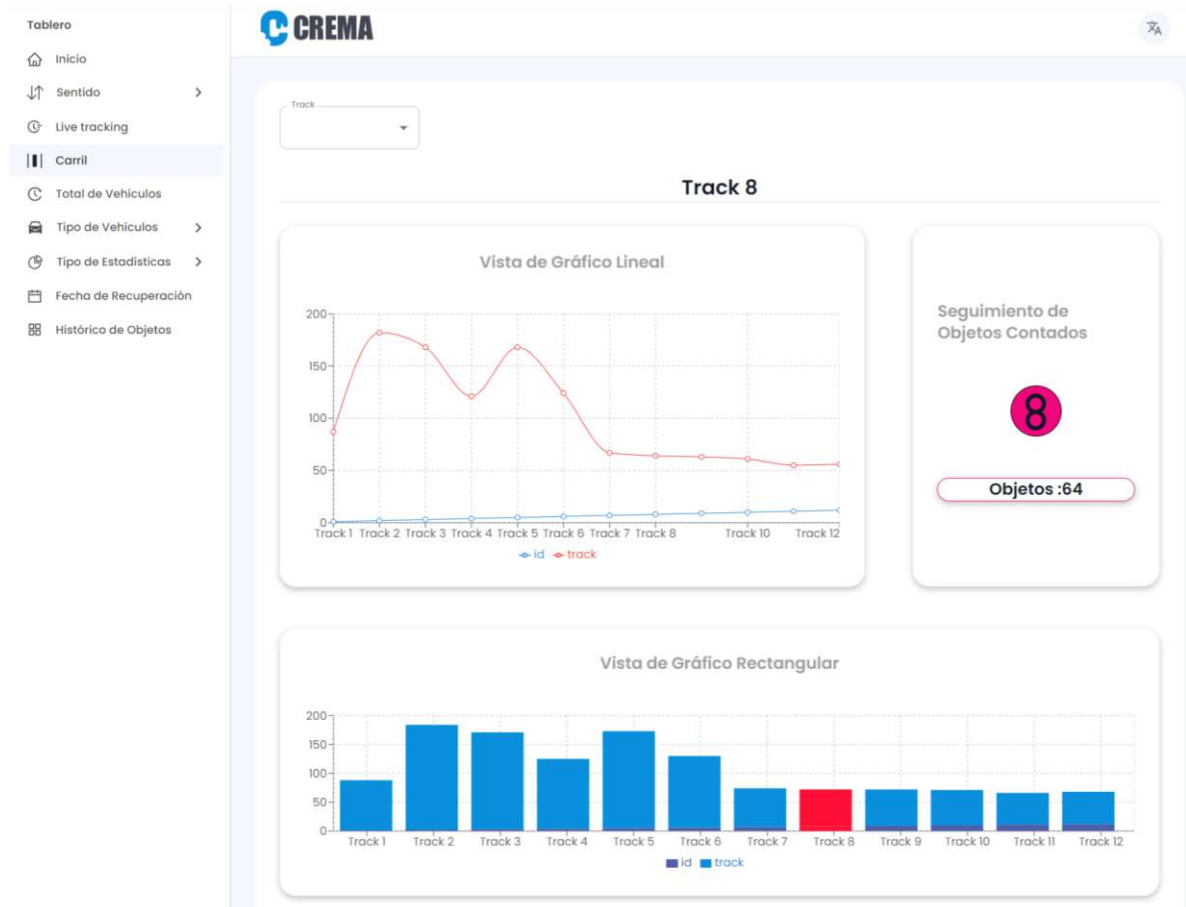
Figura 35. Carril7 sección2



Fuente: Autor

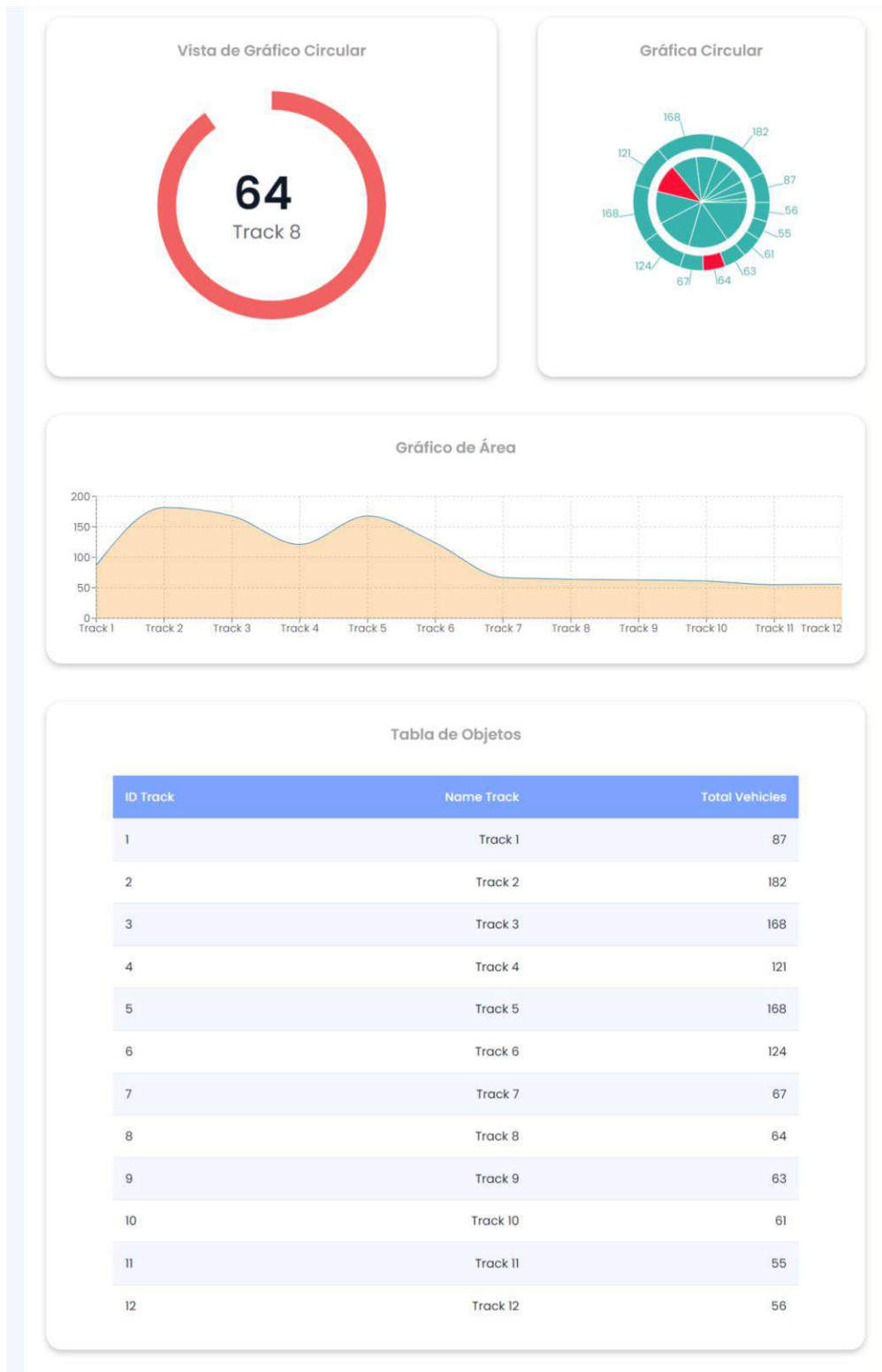
CARRIL8:

Figura 36. Carril8 sección1



Fuente: Autor

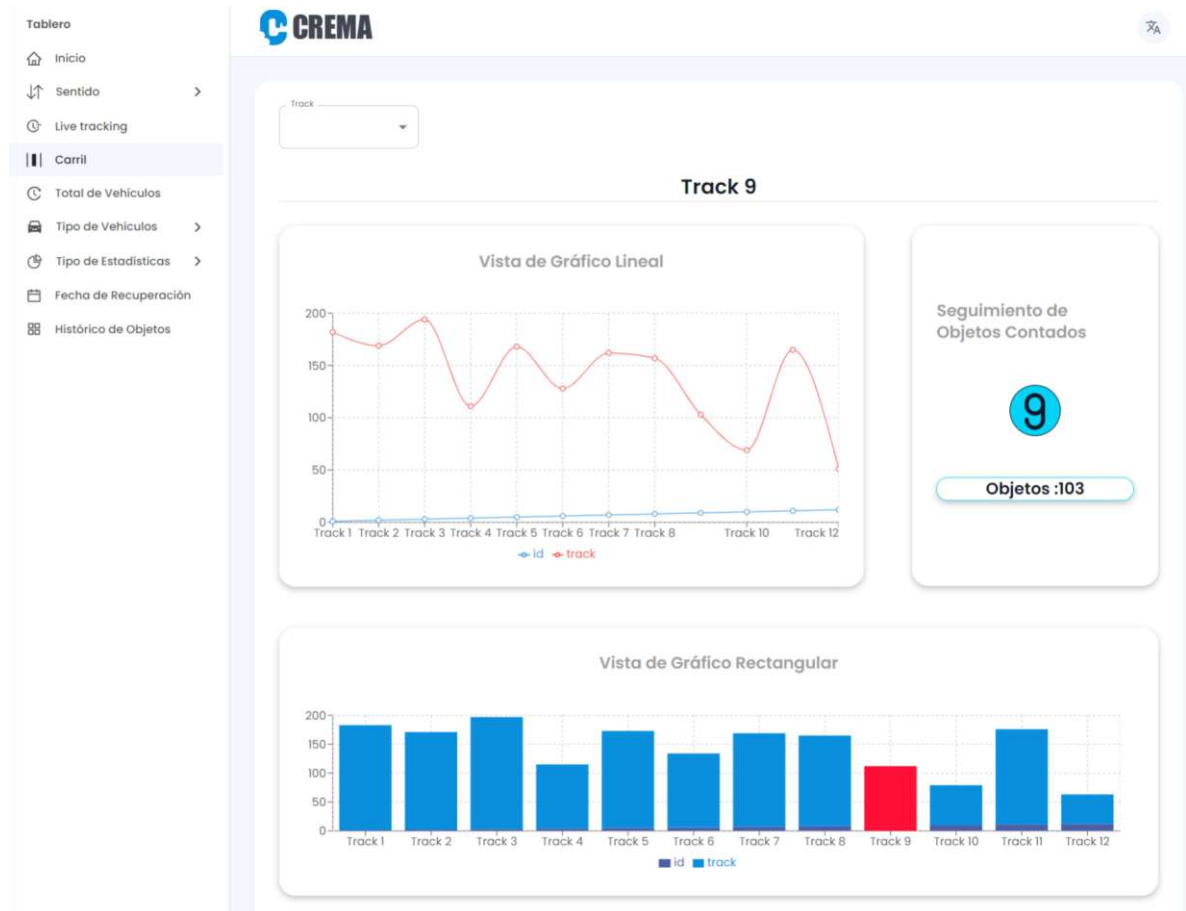
Figura 37. Carril8 sección2



Fuente: Autor

CARRIL9:

Figura 38. Carril9 sección1



Fuente: Autor

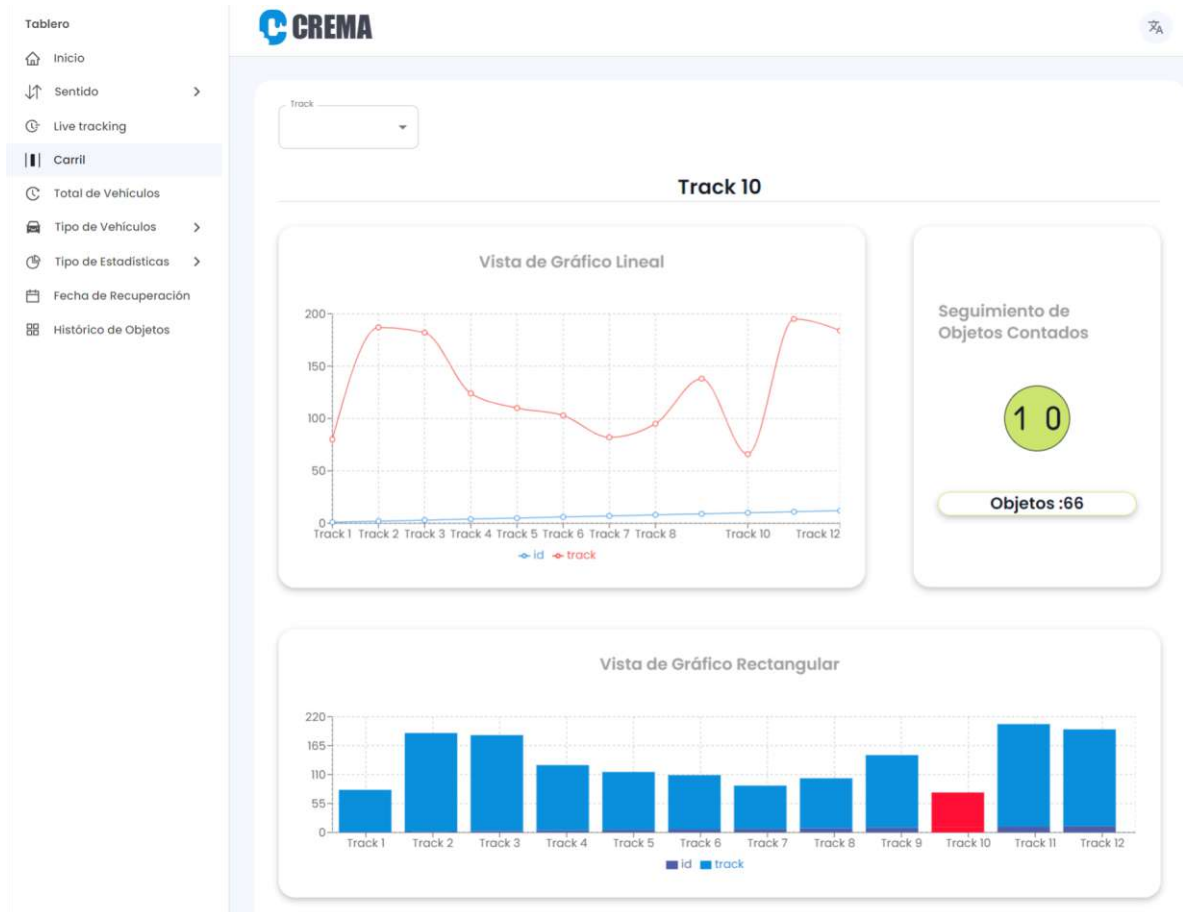
Figura 39. Carril9 sección2



Fuente: Autor

CARRIL10:

Figura 40. Carril10 sección1



Fuente: Autor

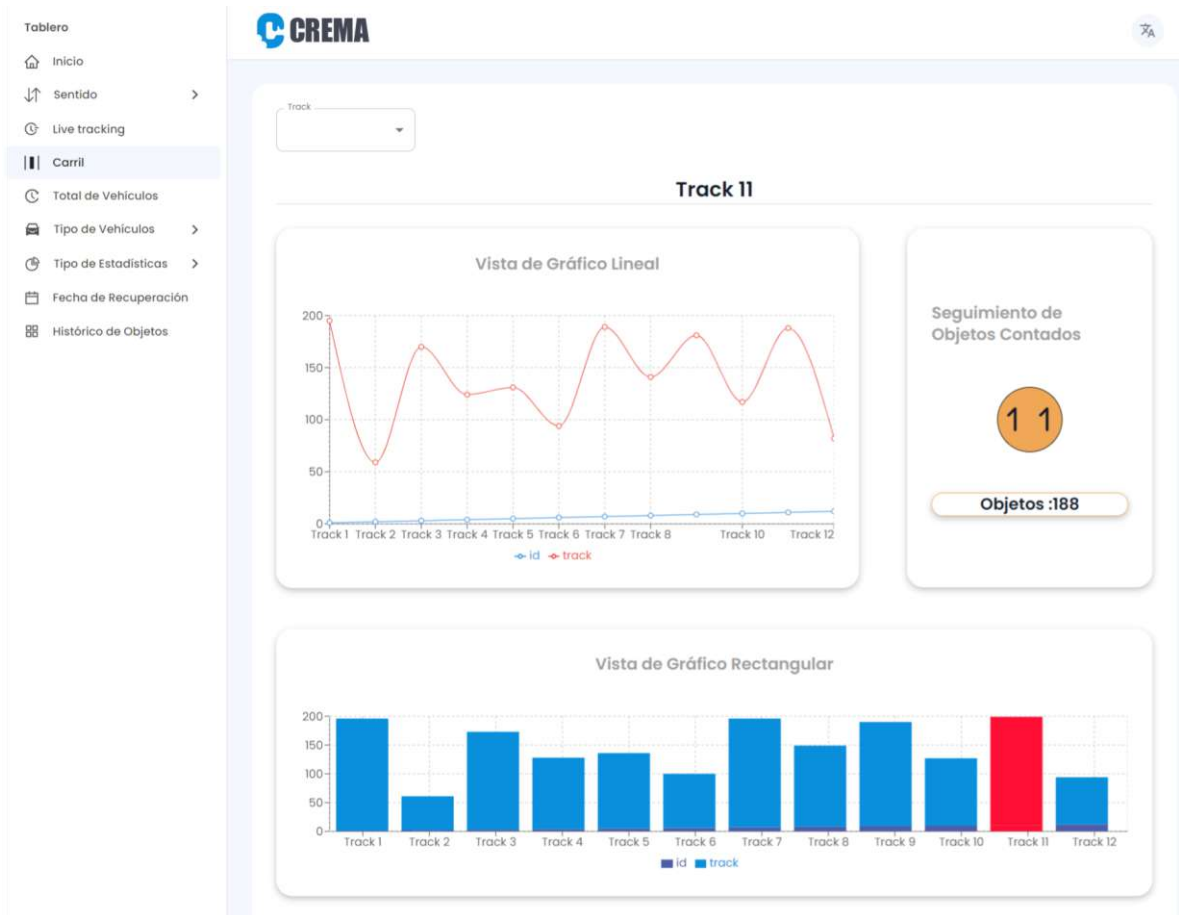
Figura 41. Carril10 sección2



Fuente: Autor

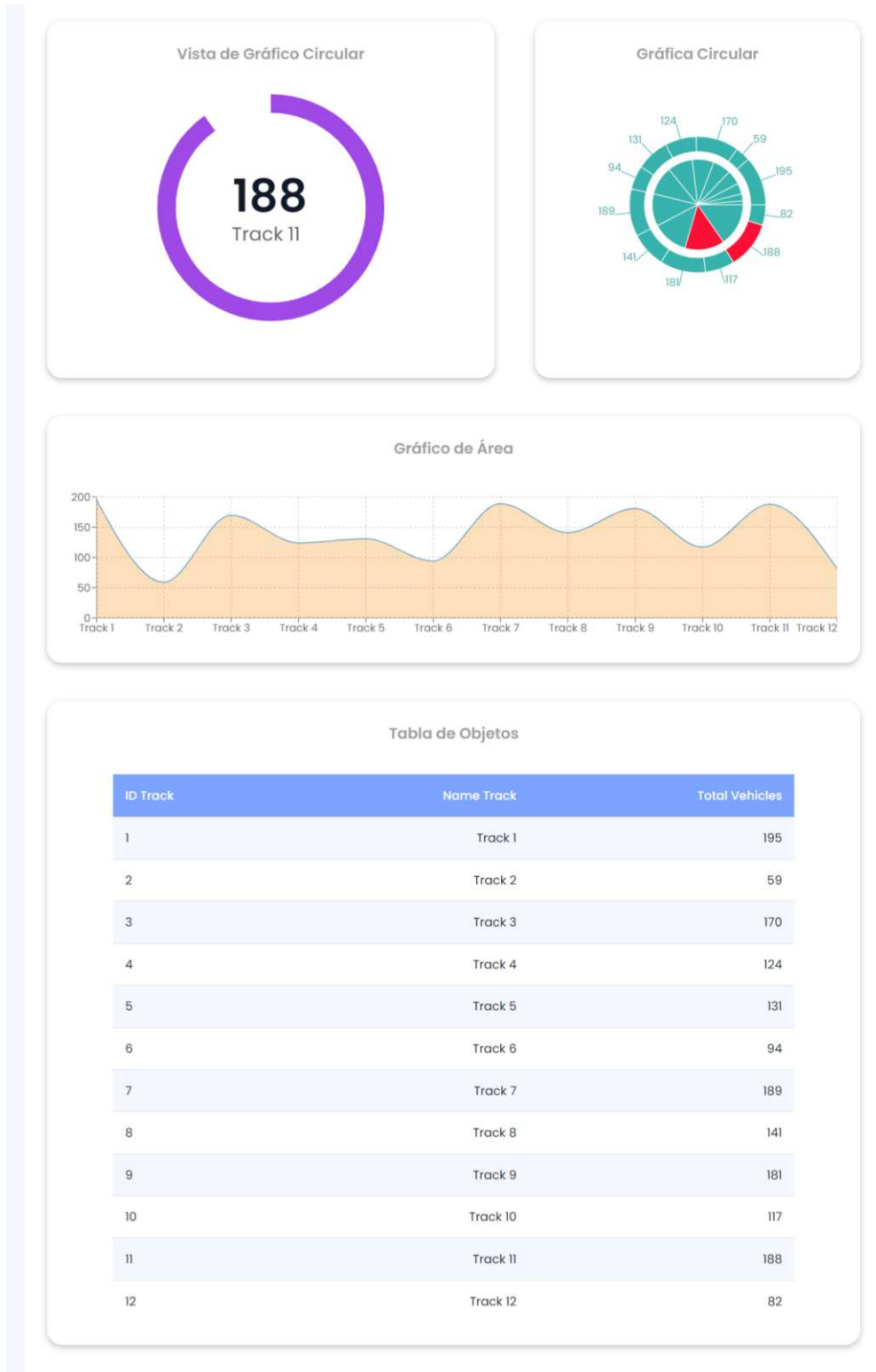
CARRIL11:

Figura 42. Carril11 sección1



Fuente: Autor

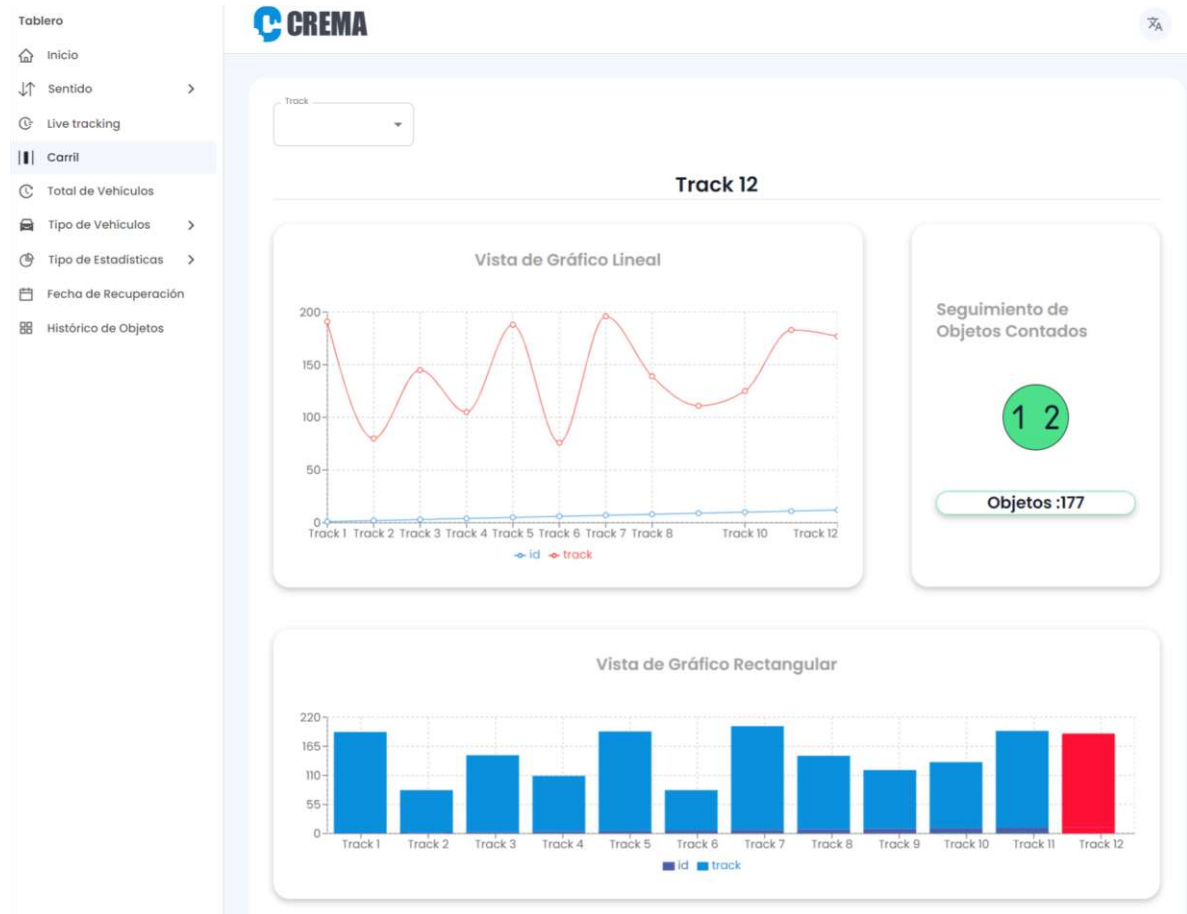
Figura 43. Carril11 sección2



Fuente: Autor

CARRIL12:

Figura 44. Carril12 sección1



Fuente: Autor

Figura 45. Carril12 sección2



Fuente: Autor

## TOTAL DE VEHÍCULOS

En la página “total de vehículos” se muestran datos correspondientes al total de vehículos que han circulado recientemente. En la primera parte se encuentra una sumatoria de total de vehículos en un círculo de progreso, junto a ítems de vehículos contados por clase; luego se encuentra una gráfica de área, seguido de una tabla de objetos donde se encuentran mencionados todos los vehículos; por último, tenemos todos los vehículos contados por clase.

Figura 46. Total de vehículos sección1



Fuente: Autor

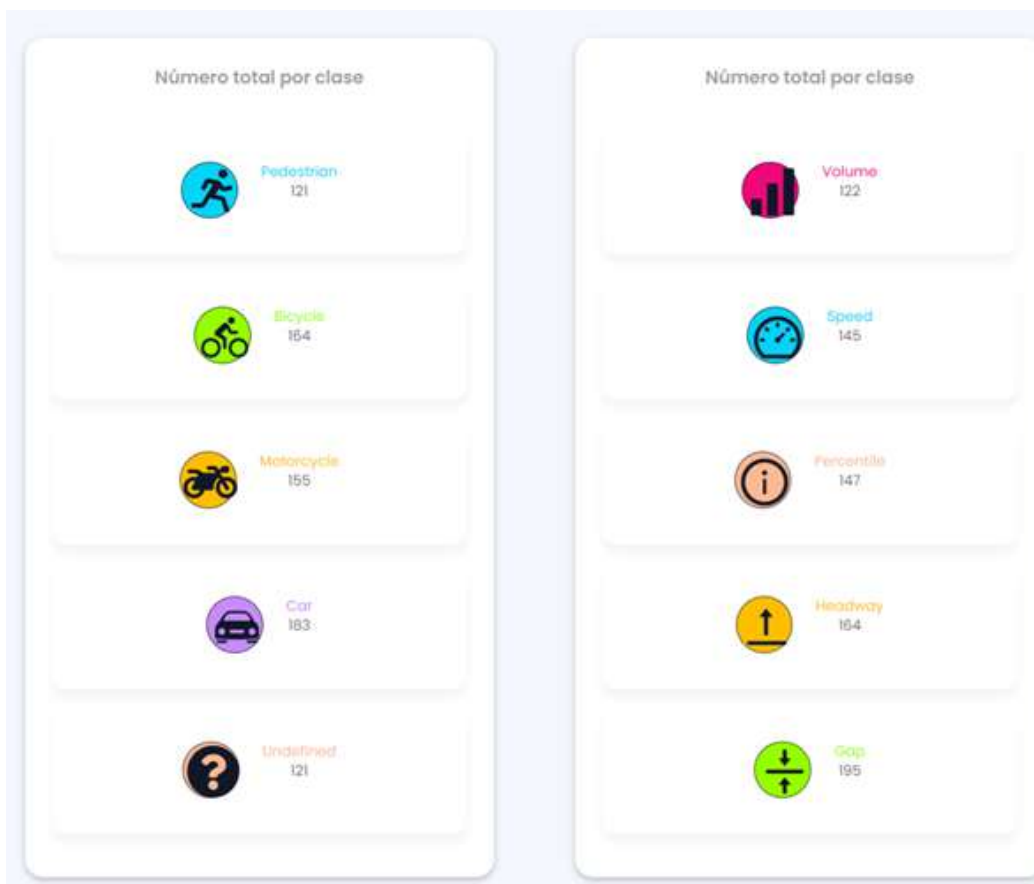
Figura 47. Total de vehículos seccion2

Tabla de Objetos

ID Track	Name Vehicles	Total Vehicles
1	Pedestrian	121
2	Bicycle	164
3	Motorcycle	155
4	Car	183
5	Short Truck	194
6	Long Truck	190
7	Undefined	121
8	Volume	122
9	Speed	145
10	Percentile	147
11	Headway	164
12	Gap	195

Fuente: Autor

Figura 48. Total de vehículos sección3



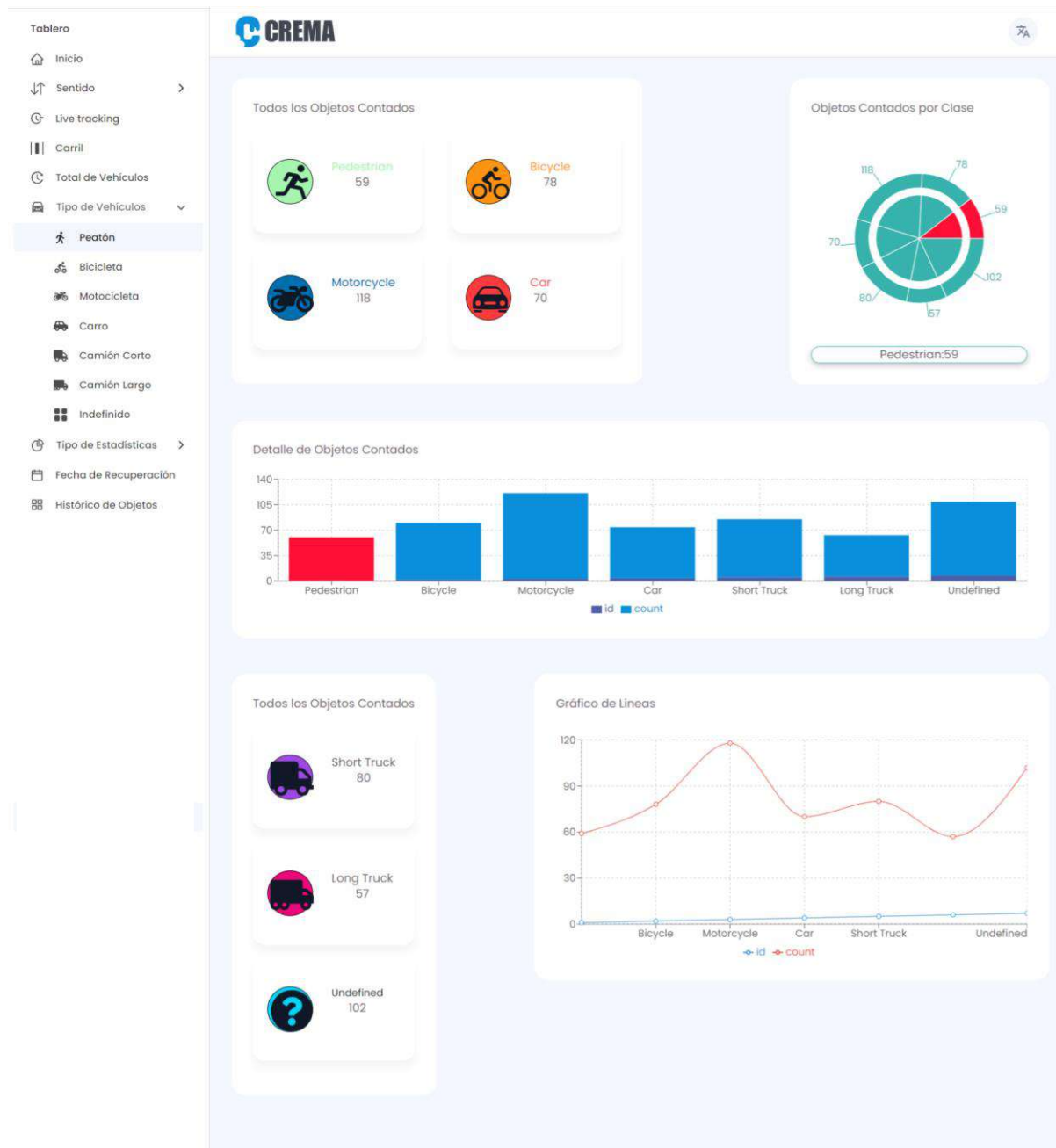
Fuente: Autor

### **TIPO DE VEHICULOS**

El ítem tipo de vehículos se encuentra en el menú lateral, el cual es desplegable y contiene varios subítems los cuales son: Peatón, bicicleta, motocicleta, carro, camión corto, camión largo e indefinido. Cada uno de esos subítems redirige a una página independiente con sus respectivos datos, las cual tiene la siguientes estructura: En primer lugar se encuentra el bloque de objetos contados por clase; seguidamente está la gráfica circular, donde se pinta de rojo únicamente los datos que corresponden a la página; luego se encuentra una gráfica rectangular, donde también están pintados de rojo los datos correspondientes a la página; por último, se encuentra otro bloque de objetos contados por clase y al lado un gráfico de líneas.

PEATÓN:

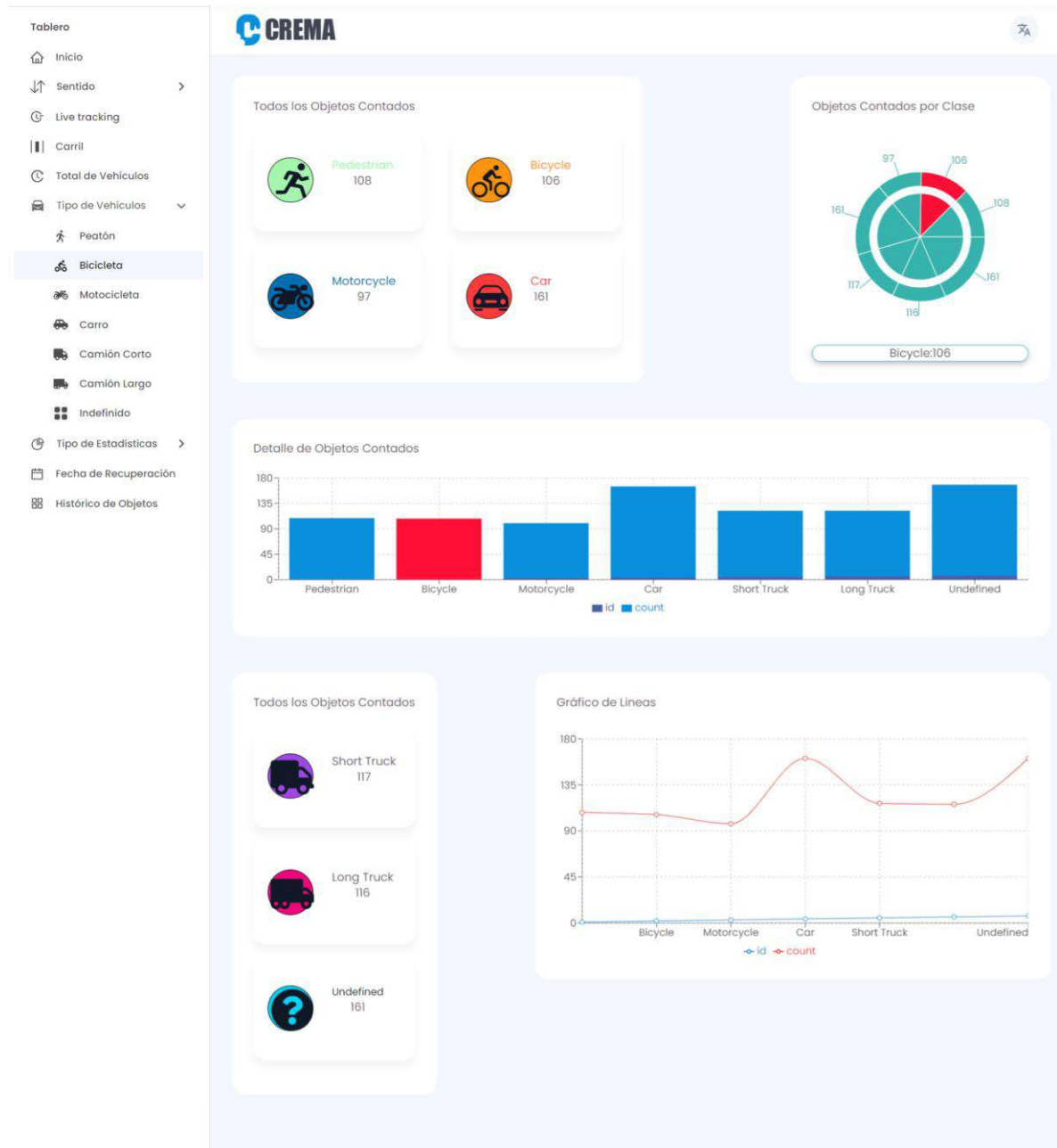
Figura 49. Peatón



Fuente: Autor

BICICLETA:

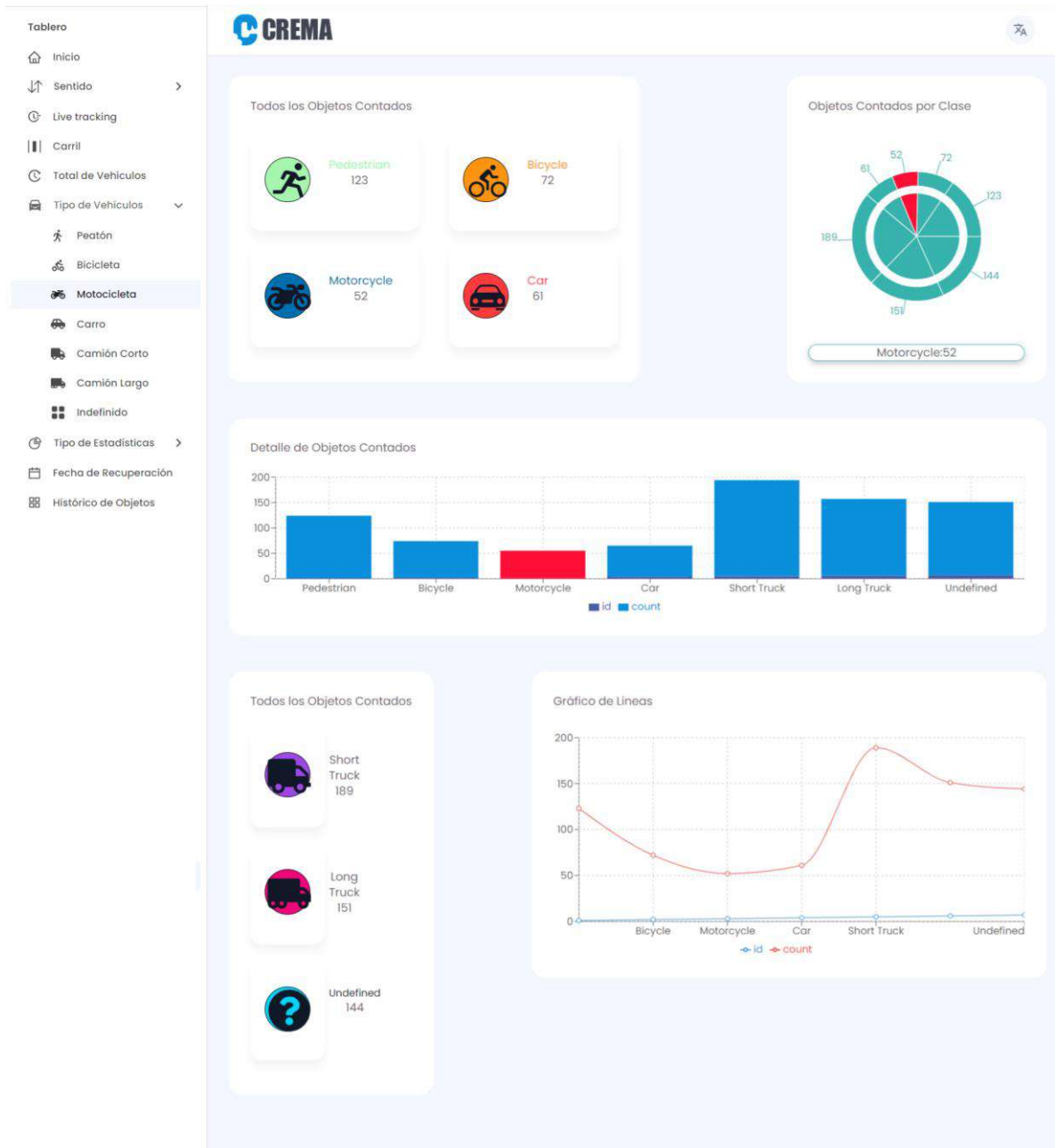
Figura 50. Bicicleta



Fuente: Autor

MOTOCICLETA:

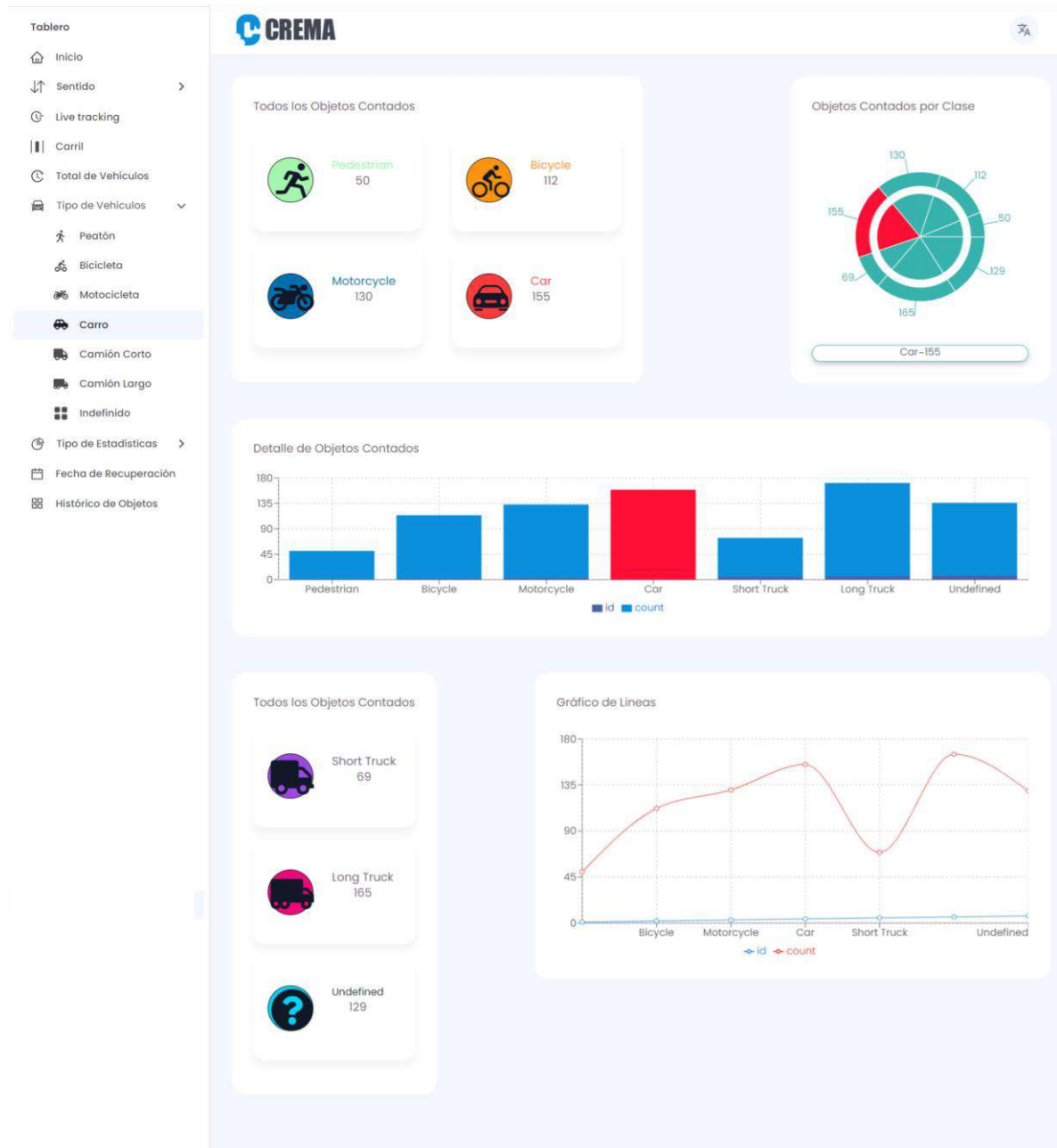
Figura 51. Motocicleta



Fuente: Autor

CARRO:

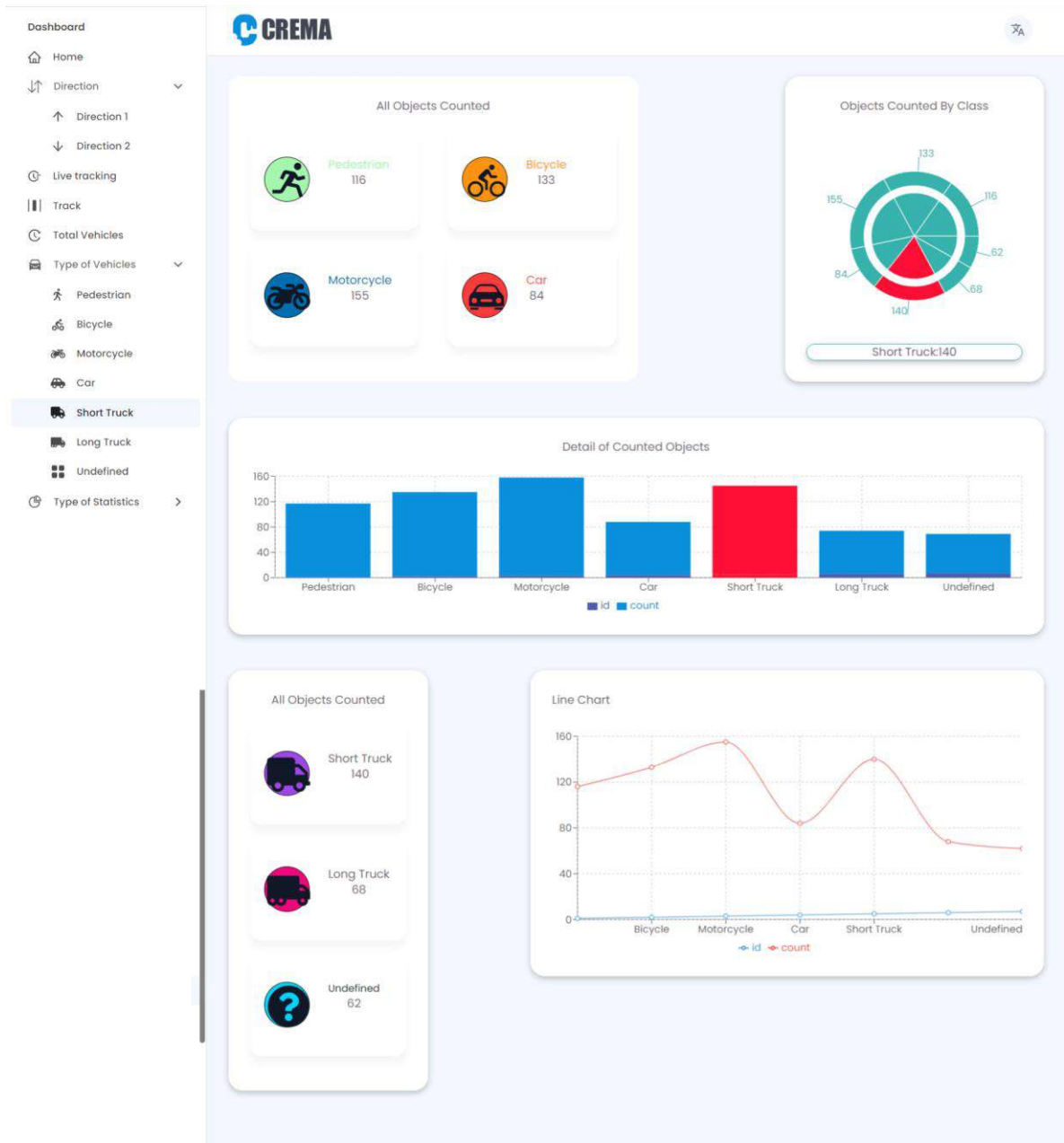
Figura 52. Carro



Fuente: Autor

CAMIÓN CORTO:

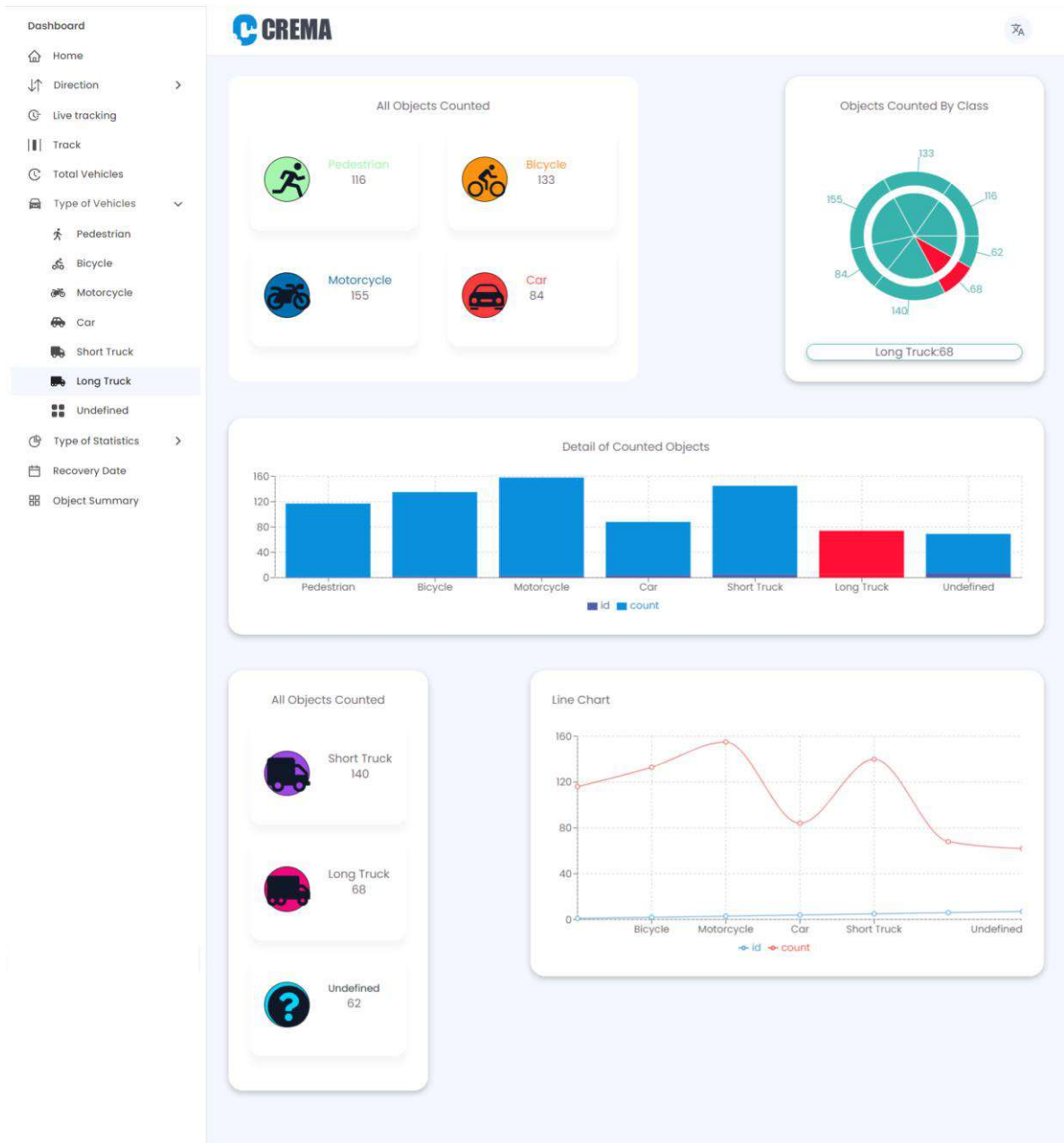
Figura 53. Camión corto



Fuente: Autor

CAMIÓN LARGO:

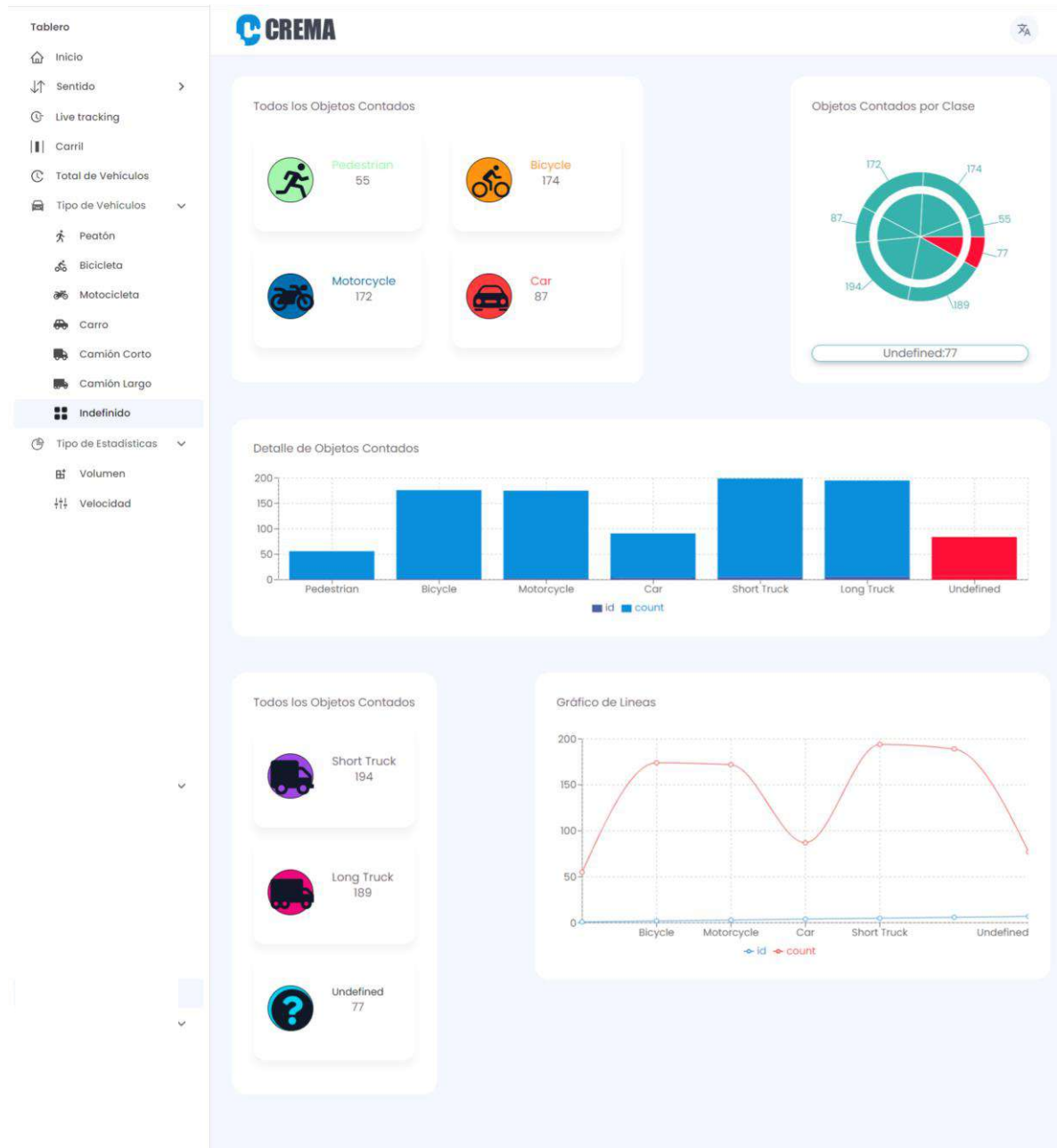
Figura 54. Camión largo



Fuente: Autor

INDEFINIDO:

Figura 55. Indefinido



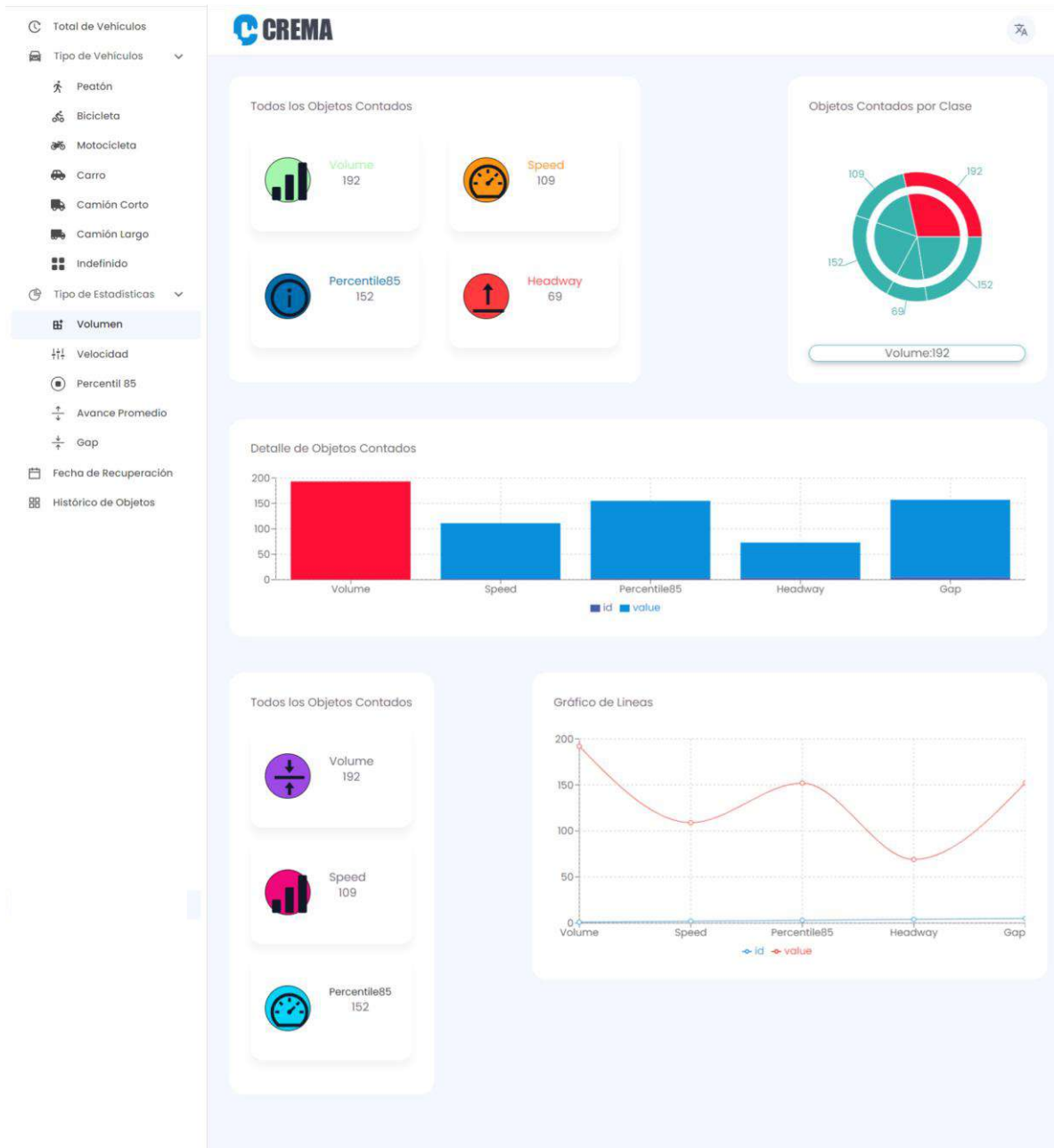
Fuente: Autor

### **TIPO DE ESTADÍSTICAS**

El ítem tipo de vehículos se encuentra en el menú lateral, el cual es desplegable y contiene varios subítems los cuales son: Peatón, bicicleta, motocicleta, carro, camión corto, camión largo e indefinido. Cada uno de esos subítems redirige a una página independiente con sus respectivos datos, las cual tiene la siguientes estructura: En primer lugar se encuentra el bloque de objetos contados por clase; seguidamente está la gráfica circular, donde se pinta de rojo únicamente los datos que corresponden a la página; luego se encuentra una gráfica rectangular, donde también están pintados de rojo los datos correspondientes a la página, por último se encuentra otro bloque de objetos contados por clase y un gráfico de líneas.

VOLUMEN:

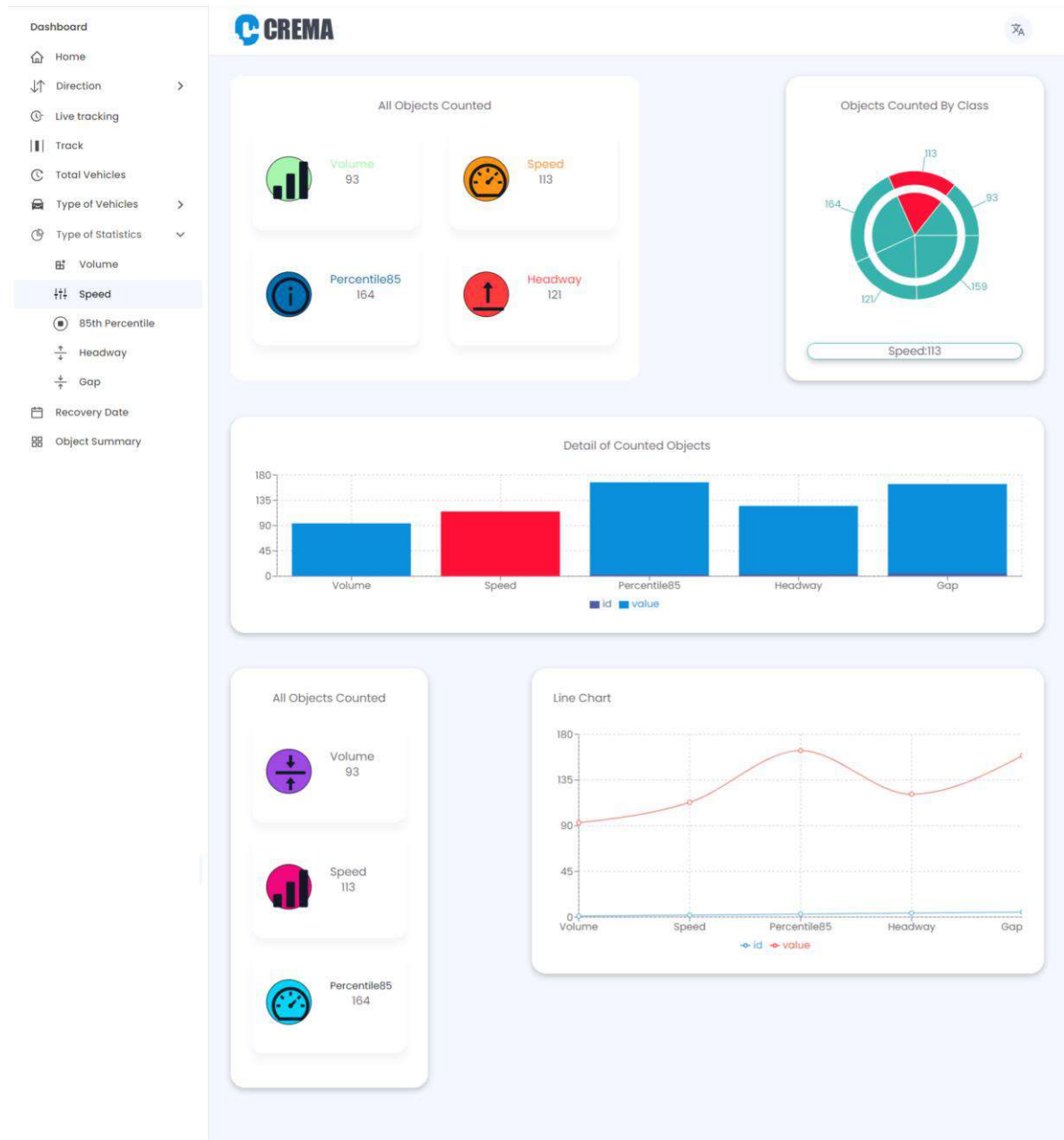
Figura 56. Volumen



Fuente: Autor

VELOCIDAD:

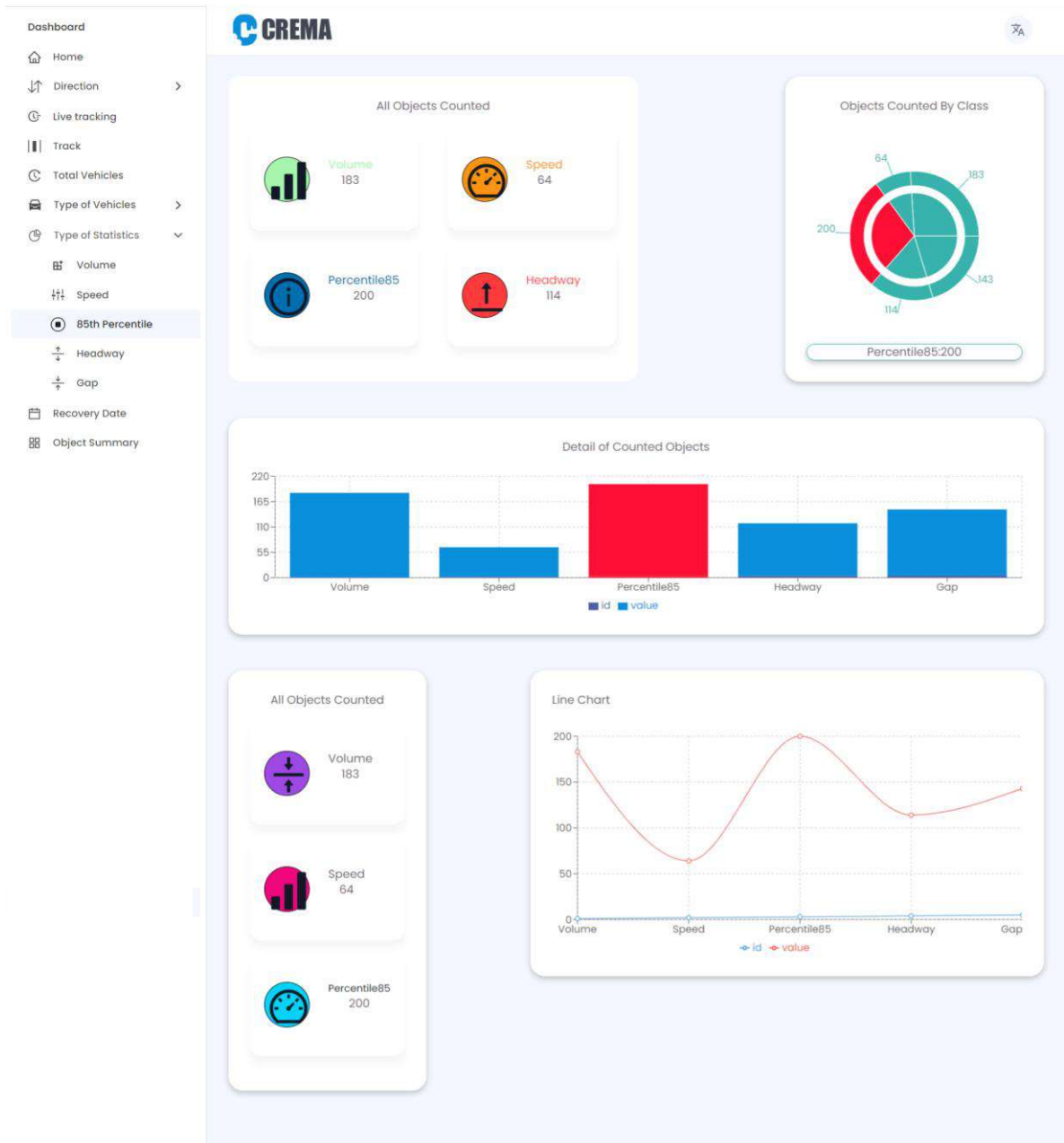
Figura 57. Velocidad



Fuente: Autor

PERCENTIL 85:

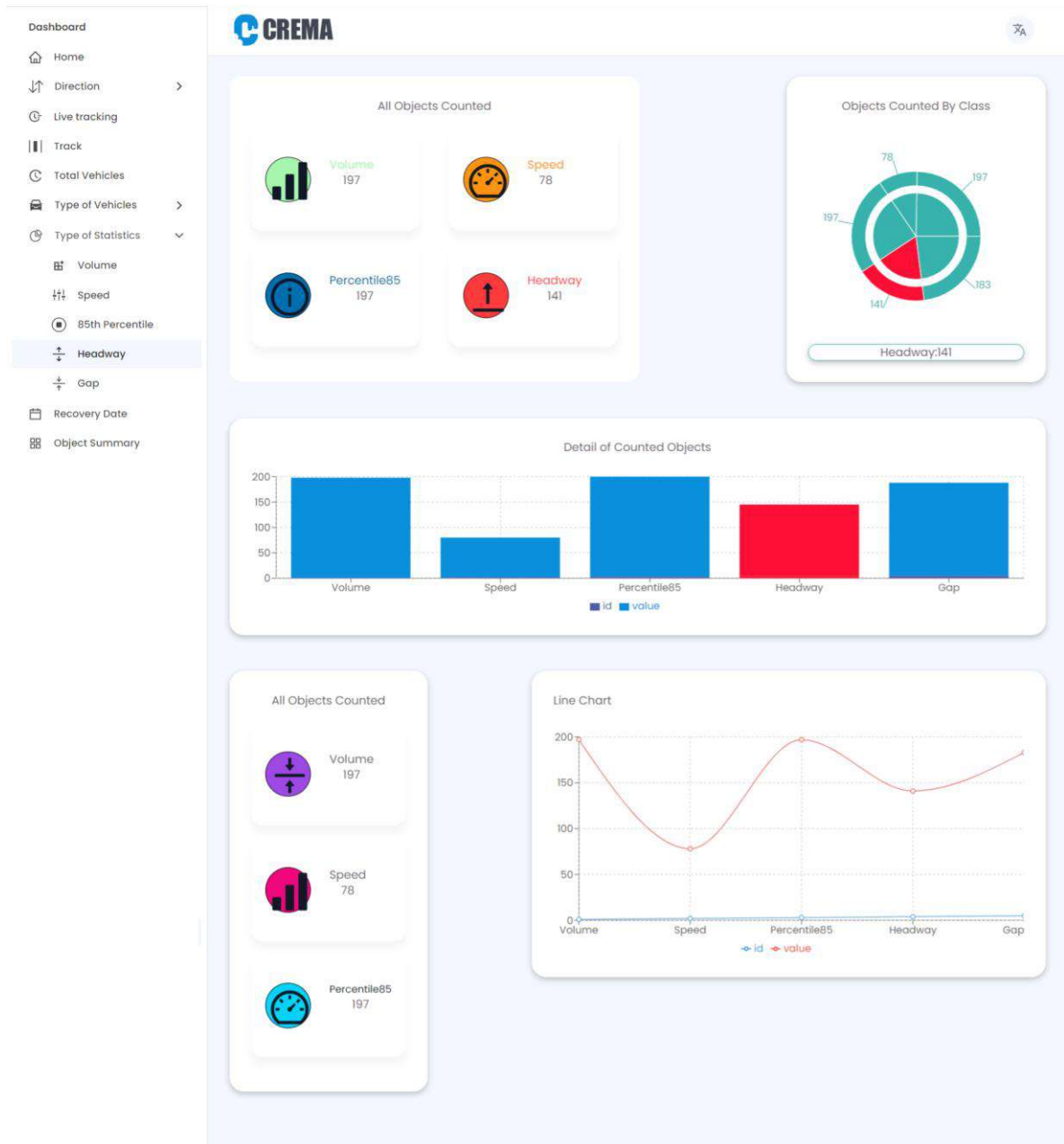
Figura 58. Percentil



Fuente: Autor

AVANCE PROMEDIO:

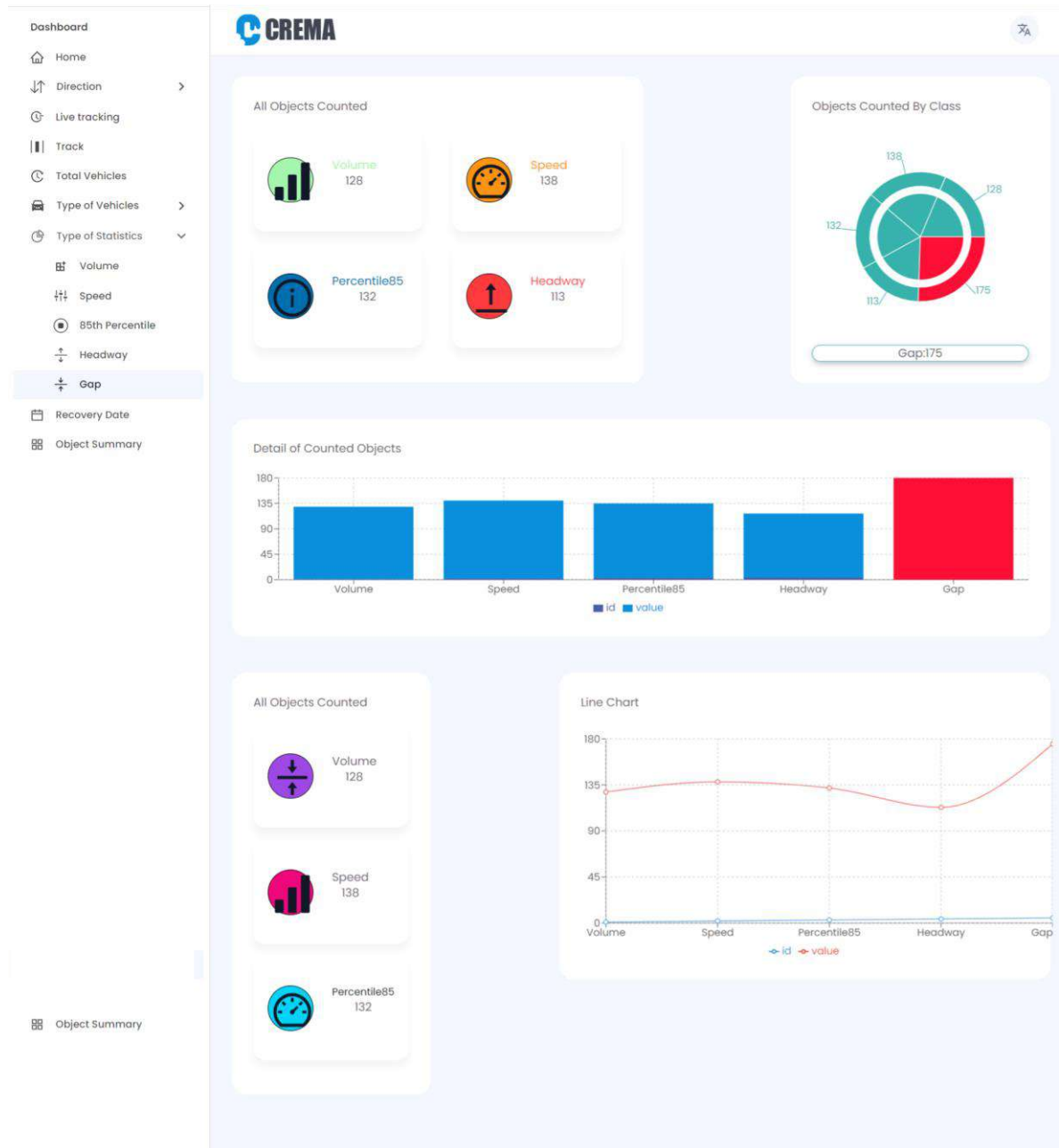
Figura 59. Avance promedio



Fuente: Autor

GAP:

Figura 60. Gap



Fuente: Autor

## FECHA DE RECUPERACIÓN

El ítem Fecha de recuperación se encuentra en el menú lateral, en esta página se observa como único elemento una tabla, en la cual se muestra la fecha en la que se ingresa al dashboard.

Figura 61. Fecha de recuperación



The screenshot shows the CREMA dashboard interface. On the left is a sidebar menu with the following items: Dashboard, Home, Direction, Live tracking, Track, Total Vehicles, Type of Vehicles, Type of Statistics, Recovery Date (highlighted), and Object Summary. The main content area is titled 'Recovery Date' and contains a table with the following data:

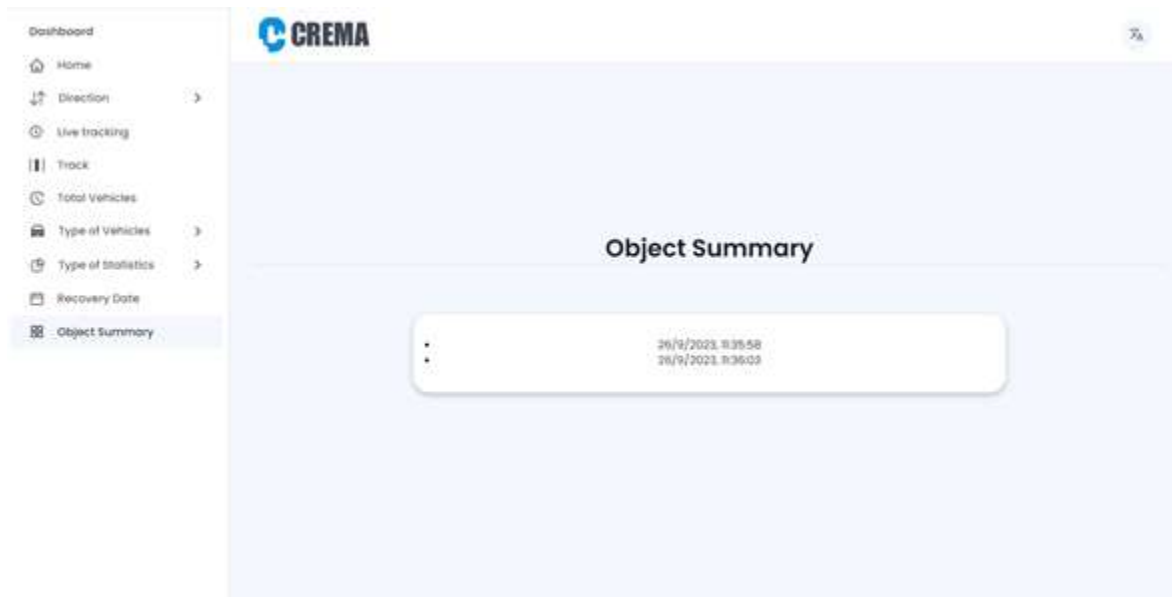
ID	Date
1	8/26/2023

Fuente: Autor

## HISTÓRICO DE OBJETOS

El ítem de histórico de objetos se encuentra ubicado en el menú lateral, esta página tiene como único elemento una lista de fechas, las cuales corresponden a los últimos ingresos realizados por el usuario al dashboard. Es importante señalar que, primero se debe acceder a la página home para seguidamente ver reflejada la fecha y hora en la página histórico de objetos.

Figura 62. Histórico de objetos



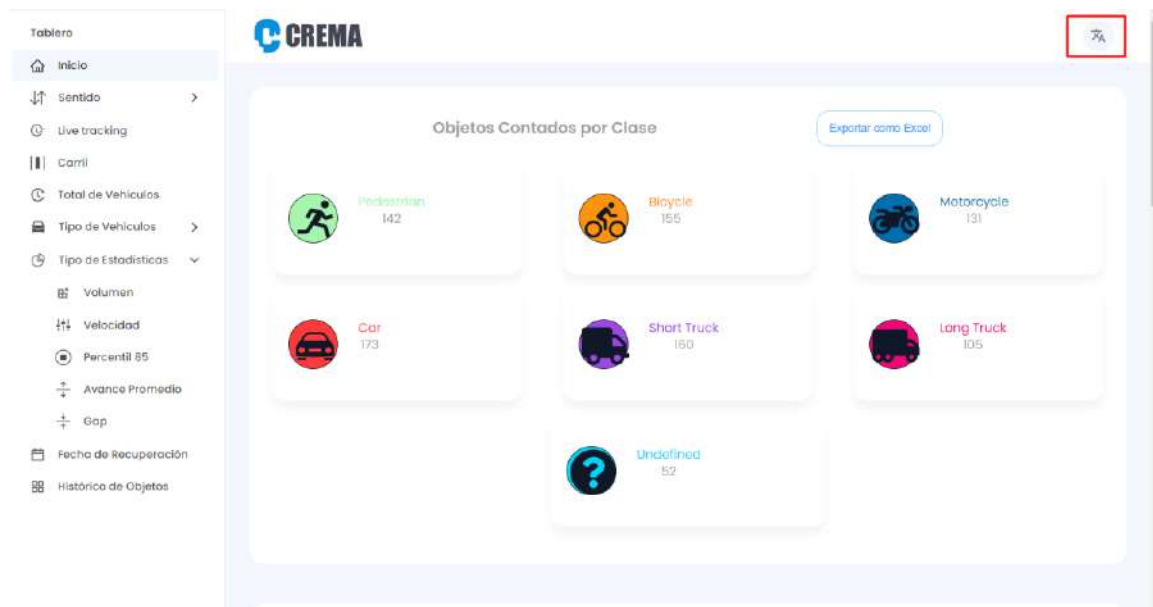
Fuente: Autor

## TRADUCCIONES

Cada una de las páginas contiene traducciones en 2 idiomas: inglés y español. Estas traducciones están aplicadas a textos tales como los ítems del menú y títulos; algunos textos correspondientes a descripciones no se encuentran traducidos, puesto que mantienen el idioma del proveedor. El botón de traducción está ubicado en la parte superior derecha. A continuación, se muestra como ejemplo la página home primeramente en idioma inglés, y después en idioma español.

ESPAÑOL:

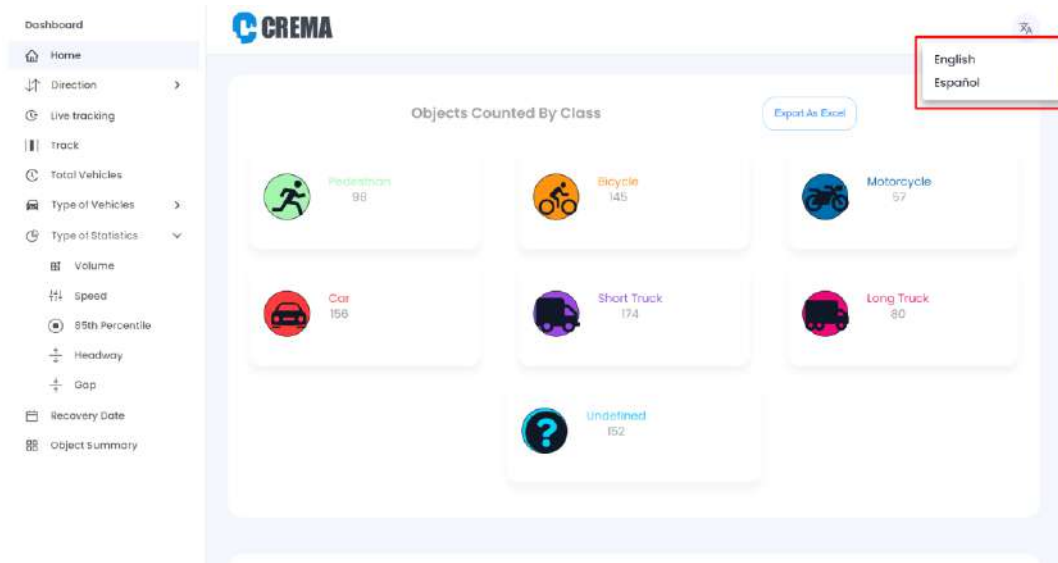
Figura 63. Idioma español



Fuente: Autor

INGLÉS:

Figura 64. Idioma ingles



Fuente: Autor

### Anexo 3. Especificación de Requerimientos

Lo requerimientos fueron informados como solicitudes hechas por el cliente, y de las cuales se desprende la elaboración y gestión del backlog. El listado de requisitos quedo de la siguiente manera:

Id. REQ	Descripción corta	Tipo (F, NF, TC)
001	El dashboard debe ser intuitivo y agradable al usuario	NF
002	El dashboard debe tener graficas	NF
003	Creación del Módulo correspondiente a Tipos de vehículos.	F
004	Creación del Módulo correspondiente a Tipos de estadísticas.	F
005	Creación de Módulos correspondientes a Otras características	F
006	Creación de Módulos correspondientes a fecha de recuperación e histórico de objetos	F
007	Creación de Módulos correspondientes a botón exportar	F
008	Creación de Módulos correspondientes a traducciones	F

F: Requisito Funcional    NF: Requisito No Funcional    TC: Requisito Técnico

Número de requisito	001
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	El dashboard debe ser intuitivo y agradable al usuario
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El dashboard debe ser intuitivo, agradable al usuario y debe seguir la misma línea de apariencia que la plantilla Crema.

Número de requisito	002
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	El dashboard debe tener gráficas
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El dashboard debe tener gráficas para gestionar mejor la información, permitiendo agilizar propósitos de análisis de datos propios de la empresa

Número de requisito	003
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación del Módulo correspondiente a Tipos de vehículos.
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El módulo correspondiente a tipo de vehículos tiene subitems que corresponden a: peatón, bicicleta, motocicleta, carro, camión corto, camión largo, indefinido

Número de requisito	004
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación del Módulo correspondiente a Tipos de estadísticas.
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El módulo correspondiente a tipo de estadísticas tiene subitems que corresponden a: volumen, velocidad, percentil 85, avance promedio, gap

Número de requisito	005
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación del Módulo correspondiente a Otras características.
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El módulo correspondiente a Otras características tiene subitems que corresponden a: Home, Dirección, Live tracking, Carriles y Total de vehículos

Número de requisito	006
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación de Módulos correspondientes a fecha de recuperación e histórico de objetos
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	Se deben tener en cuentas fechas de visualización del dashboard

Número de requisito	007
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación de Módulos correspondientes a botón exportar
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	El módulo correspondiente al botón de exportar datos, debe corroborar que los datos se exporten a un archivo .xlsx

Número de requisito	008
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Creación de Módulos correspondientes a traducciones
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción del requisito	Se pueden hacer traducciones tanto a ingles como español

#### **Anexo 4. Implementación metodología SCRUM**

En este apartado se exponen las herramientas que se manejaron durante el proceso de construcción del proyecto, y que a su vez como bien se indica, fueron utilizadas con la finalidad de hacer implementación de la metodología SCRUM para agilizar y facilitar la culminación del proyecto

#### **ROLES**

Primeramente, se indica que los roles que se contemplan dentro de la metodología SCRUM tales como Product Owner, Scrum Master y Scrum Team fueron asignados de la siguiente manera:

Product Owner: Jorge Rodríguez - Project manager de la empresa Roch Electronics

Funciones: Se encargó de la coordinación de tareas siempre con un rol de liderazgo y hacer seguimiento del progreso del proyecto.

Scrum Master: Nicolas Rodríguez

Funciones: Fue mentor dentro del proceso de construcción del proyecto y además ayudó como intermediario con el cliente.

Scrum Team: Ericka Sora

Funciones: Hacer desarrollo del proyecto.

## BACKLOG

Seguidamente, se presenta la lista de tareas o Backlog que se hizo y que a su vez fueron culminadas satisfactoriamente. Estas tareas se manejaron con la plataforma Trello para gestionar que el orden de la ejecución de las mismas sea el correcto y que todas sean culminadas satisfactoriamente.



## SPRINT

Así mismo, se hizo la planificación y ejecución de los sprint con una duración mensual, los cuales contaron con sus respectivas reuniones periódicas para control del avance del proyecto y aprobación del desarrollo. Haciendo un contraste con las tareas listadas en el backlog, los sprint que se ejecutaron en el transcurso de la construcción del proyecto se manejaron de la siguiente manera:

Fecha (2023)	Sprint	Descripción
Marzo	Sprint1	Definición de requerimientos Casos de uso Diagrama de Clases Elección de Tecnología

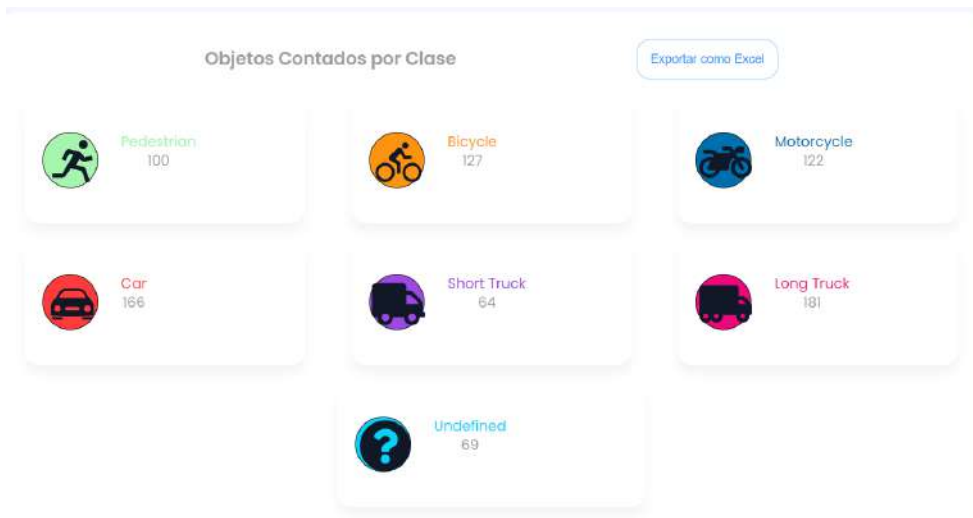
Abril	Sprint2	<ul style="list-style-type: none"> <li>*Adecuación de herramientas</li> <li>*Instalación de librerías</li> <li>*Creación de sketch para módulo de estadísticas (volumen, velocidad, percentil 85, avance promedio, gap)</li> <li>*Creación de sketch para módulo de tipo de vehículos (peatón, bicicleta, motocicleta, carro, camión corto, camión largo, indefinido)</li> <li>*Creación de sketch para módulo de Otras características (Home, Dirección, Live tracking, Carriles y Total de vehículos)</li> </ul>
Mayo	Sprint3	Montaje de información gráfica correspondiente al módulo de estadísticas (volumen, velocidad, percentil 85, avance promedio, gap)
Junio	Sprint4	Montaje de información gráfica correspondiente al módulo de tipo de vehículos (peatón, bicicleta, motocicleta, carro, camión corto, camión largo, indefinido)
Julio	Sprint5	Montaje de información gráfica correspondiente a los Módulos de Home, Dirección, live tracking, carriles, fecha de recuperación e histórico de objetos.
Agosto	Sprint6	Implementación del botón exportar
Septiembre	Sprint7	Pruebas

## Anexo 5. Pruebas Realizadas

**1.Prueba de componentes:** Comprobar que el botón Exportar como Excel, funcione correctamente

**2.Prueba de componentes:** Evidenciar que los datos descargados en el archivo de extensión .xlsx corresponden a los datos mostrados gráficamente

**Resultado:** A continuación, se hace una muestra de los datos que se muestran el dashboard en el momento de la prueba



Seguidamente, se evidencia el correcto funcionamiento de botón exportar



Y por último podemos evidenciar que los datos corresponden efectivamente a los que se estaban mostrando en pantalla

1	id	name	count						
2	1	Pedestrian	100						
3	2	Bicycle	127						
4	3	Motorcycle	122						
5	4	Car	166						
6	5	Short Truck	64						
7	6	Long Truck	181						
8	7	Undefined	69						
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									

Track **By class** By direction By statistics (+)

**3.Prueba de componentes:** Comprobar la fecha mostrada en la página Fecha de recuperación

The screenshot shows the CREMA dashboard interface. On the left, a sidebar menu lists various dashboard components, with 'Fecha de Recuperación' highlighted in a red box. The main content area is titled 'Fecha de Recuperación' and displays a table with the following data:

ID	Date
1	14/12/2023

**Resultado:** La fecha mostrada en la página fecha de recuperación coincide con la fecha en la que se está revisando el dashboard

**4.Prueba de componentes:** Constatar la lista de fechas mostradas en la página Histórico de objetos

The screenshot shows the CREMA dashboard interface. On the left, a sidebar menu lists various dashboard components, with 'Histórico de Objetos' highlighted in a red box. The main content area is titled 'Histórico De Objetos' and displays a list of two entries:

- 14/12/2023, 10:20:30
- 14/12/2023, 10:33:16

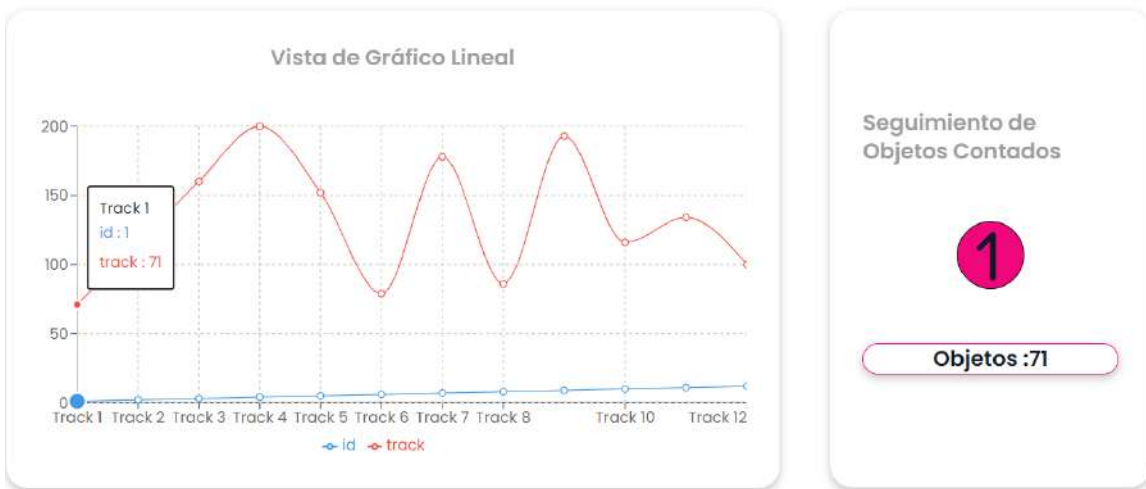
**Resultado:** La lista de fechas que se muestran en la página Histórico de objetos, coincide con las fechas en las que se ingresó al dashboard recientemente

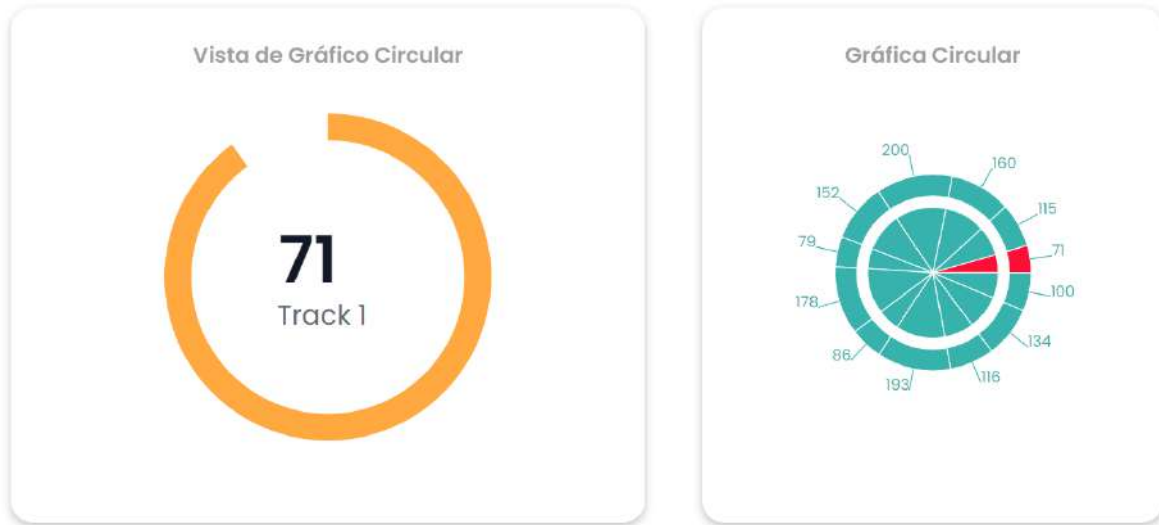
**5.Prueba de componentes:** Comprobar el seleccionador de la página Carril

Se escoge el carril1 y podemos observar que efectivamente los datos se filtran solo para el carril correspondiente



### Track 1

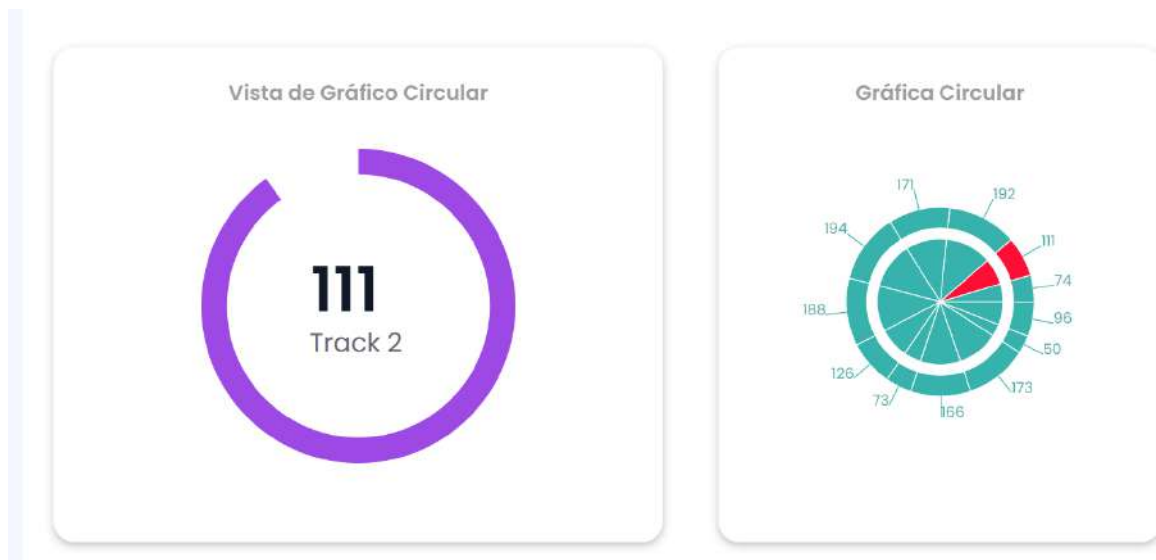




Del mismo modo se prueba en los 11 carriles restantes, pero se colocaron 2 carriles para hacer la muestra de evidencia

### Track 2



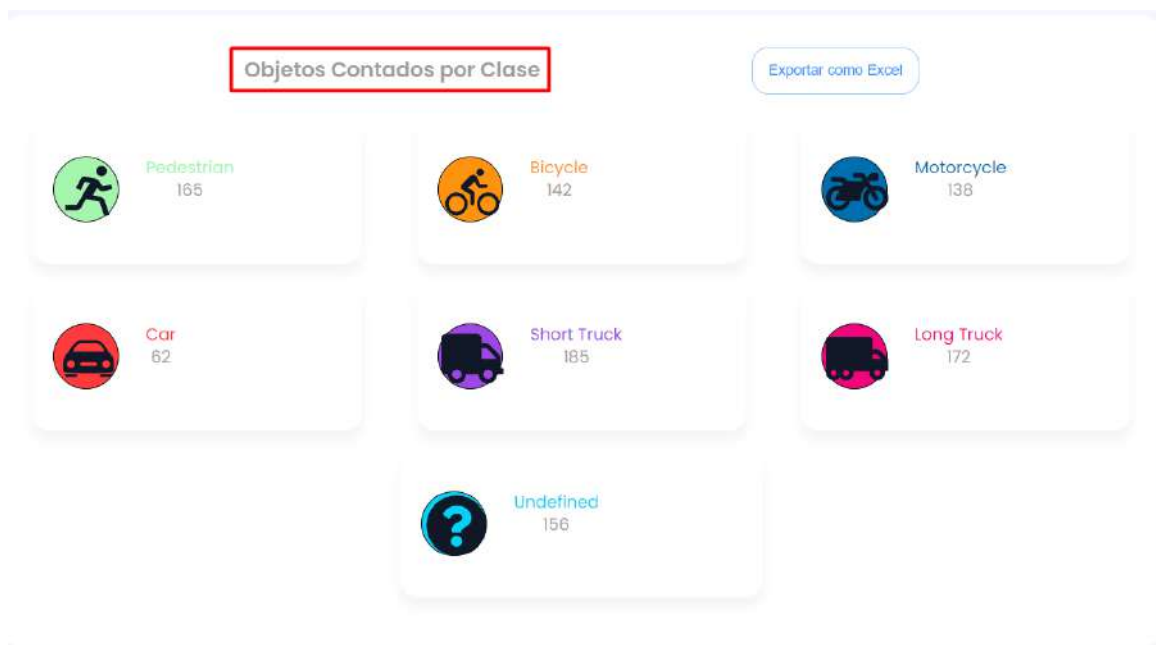


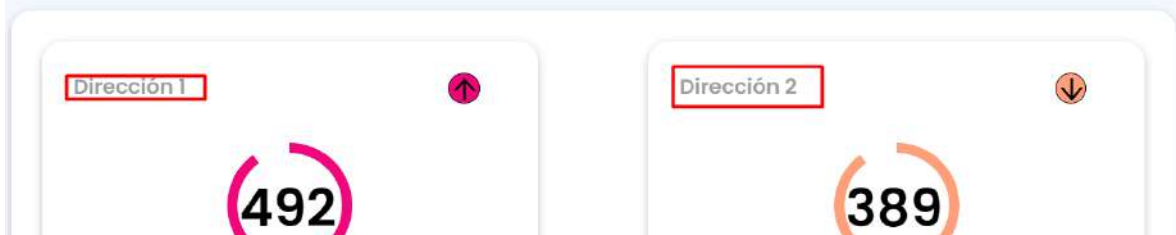
**Resultado:** Cada uno de los ítems que pertenecen al selector de la página Carril, cumplen satisfactoriamente con mostrar únicamente los datos que se piden. Siendo estos ítems enumerados desde el Carril1 hasta el carril12

**6. Prueba de integración:** Corroborar las traducciones de todas las páginas

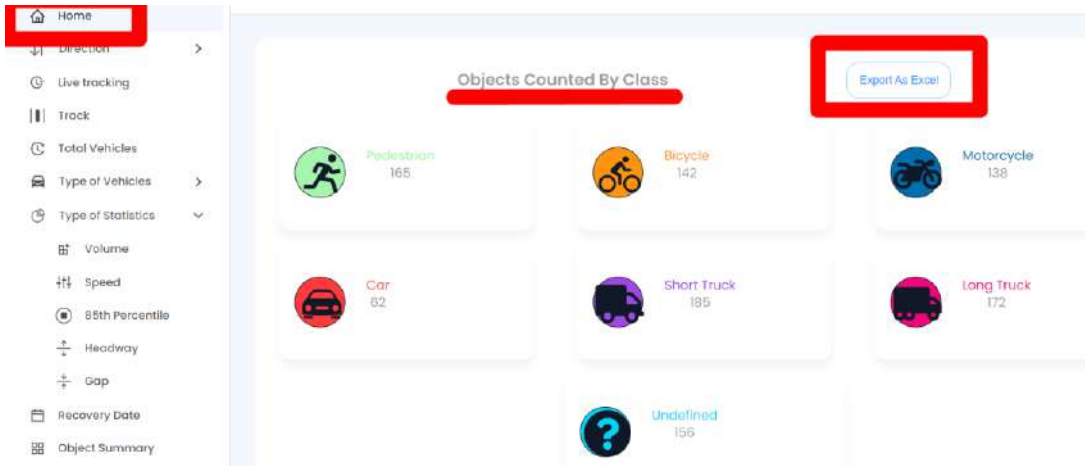
Se toma como referencia para mostrar la pagina inicio

Español



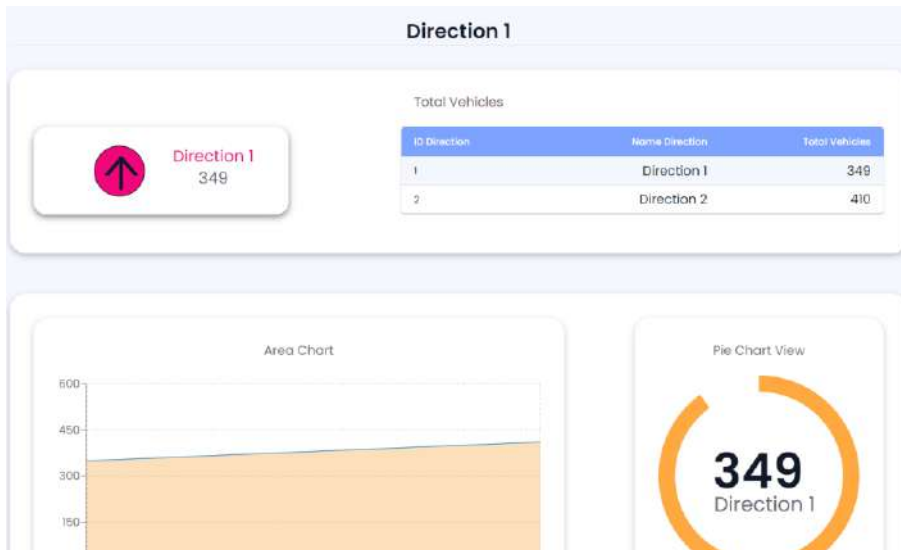


Ingles



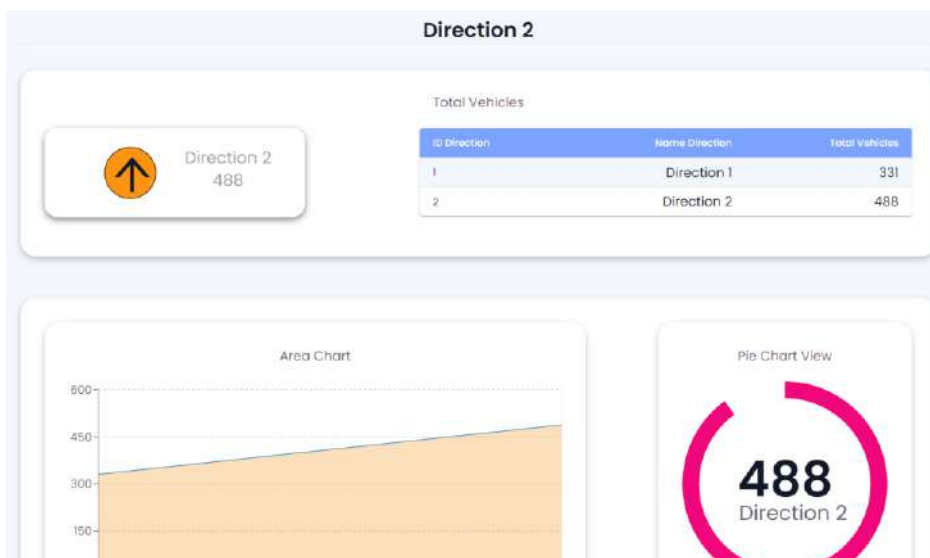
**Resultado:** Los textos como títulos y descripciones que no vienen directamente de la información suministrada, fueron traducidos satisfactoriamente

**7. Prueba de integración:** Examinar el enlace entre la API y la página de dirección 1 En la tabla del inicio se evidencia que los datos pertenecen al ítem de dirección



**Resultado:** Se evidencia que los datos se filtran correctamente y corresponden a Dirección 1

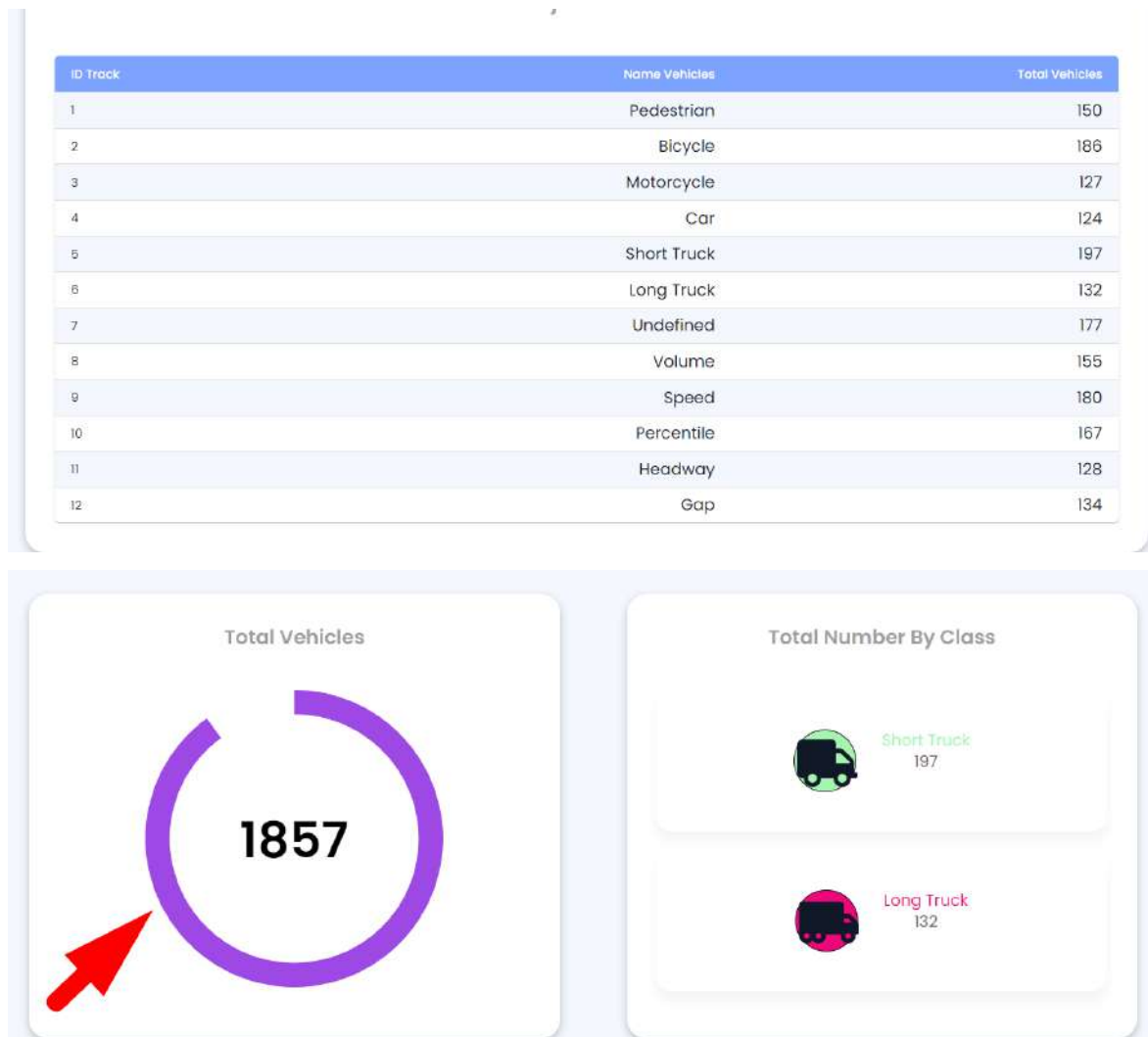
**8. Prueba de integración:** Examinar el enlace entre la API y la página de dirección 2 En la tabla del inicio se evidencia que los datos pertenecen al ítem de dirección



**Resultado:** Se evidencia que los datos se filtran correctamente y corresponden a Dirección2

**10.Prueba de integración:** Comparar el resultado de la sumatoria de todos los vehículos, con el número de vehículos mostrados en las gráficas; en la página Total de vehículos

En la tabla se evidencia que se encuentran enumerados el total de vehículos, y de allí se saca la sumatoria final que se encuentra en la segunda imagen parte superior



**Resultados:** El resultado coincide con la sumatoria de los valores echa manualmente