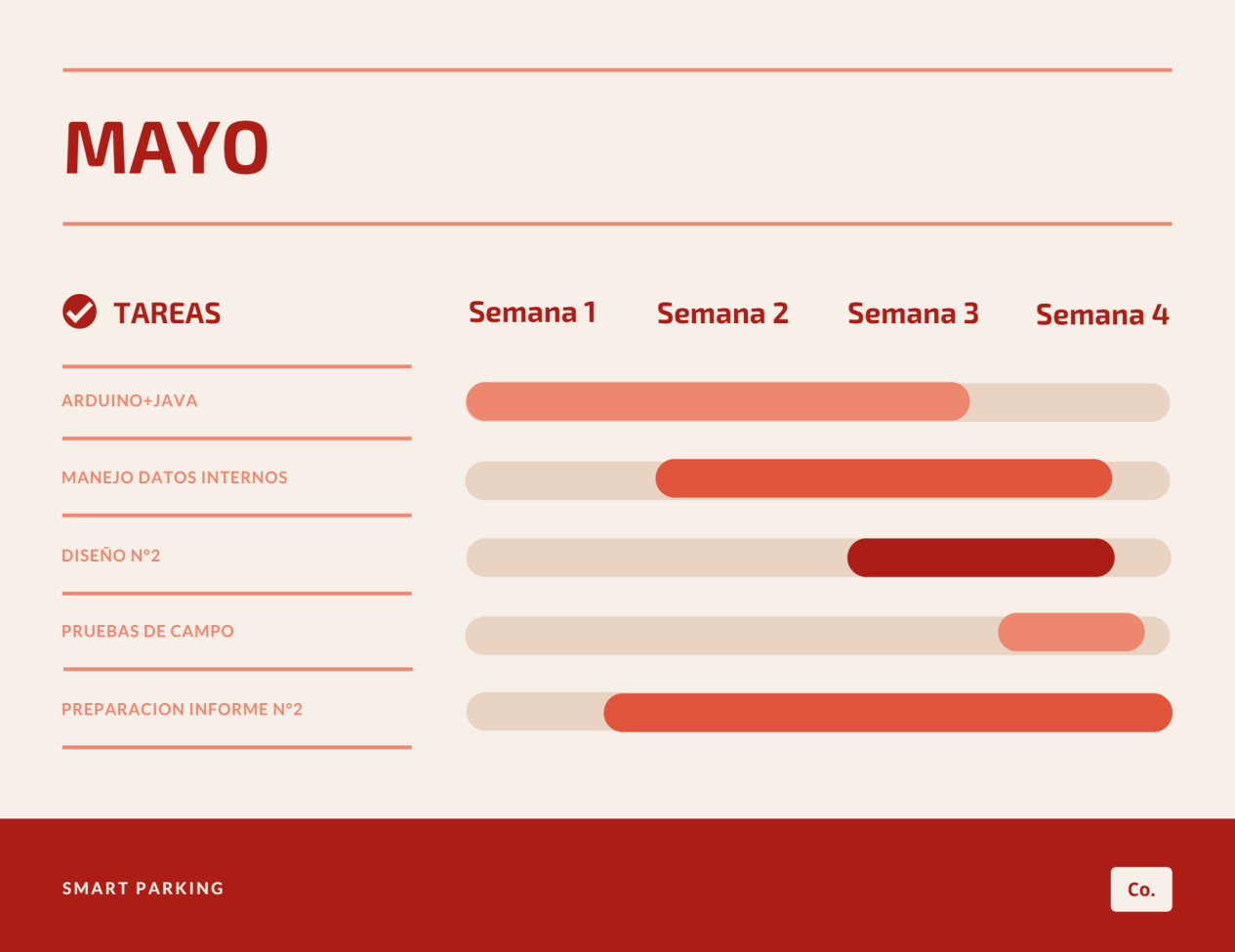
Introducción.

En el mundo existen más de 1,1 billones de vehículos, en Chile más de 5 millones, en la Araucanía más de 220 mil (datos entregados por el instituto nacional de estadística en el año 2016), con el rápido avance de las ciudades y del crecimiento demográfico los espacios para construir estacionamientos se han reducido radicalmente y la demanda de estos crece exponencialmente cada año. Por esta razón se vuelve cada vez más necesario un sistema que sea capaz de optimizar y facilitar su manejo.

Con esto en mente nuestro equipo de trabajo busca desarrollar un sistema que permita detectar y manejar el estado de los espacios de un estacionamiento y con ello facilitar su gestión, navegabilidad además de reducir el tiempo invertido en estacionar. Todo esto con el fin de optimizar el uso de este recurso cada vez más escaso en la sociedad actual.

Para este prototipo nos basamos principalmente en tecnologias de Java y Arduinos, además de herramientas de diseño, para generar las primeras propuestas de interfaces gráficas, y herramientas que permitan generar u organizar calendarios de trabajo y asignar responsabilidades. Adjunto a este informe se entrega un prototipo básico del proyecto y nuestras impresiones de este.

# *Cambios en planificación.*

La planificación sufrío cambios debido a problemas con la generación de la conexión entre arduino y java, problemas con puertos, el sistema operativo y con el manejo de la biblioteca nos obligo a trabajar 2 semanas extras en este aspecto, lo que retraso también el resto del trabajo ya que la conexión de estas tecnologias es parte principal del proyecto por lo que la nueva planificación tiene menos tiempo de trabajo en ciertos aspectos como diseño para cumplir con los plazos requeridos. 

Gestión de datos.

Como se mencionó en los cambios, se agregará una funcionalidad que guarde algunos datos relevantes.

El archivo que manejará sera del tipo txt, con 3 columnas la primera con el sensor desde el que llegarón los datos, la segunda con la fecha y hora en la qué el espacio se encuentra ocupado y la tercera columna la fecha en la que vuelve a estar disponible.

Abra una sola clase que se encargue de esto la clase ManejoDatos en el paquete manejodedatos, para ello se generarán cambios en la clase lote, además de un Inferface y un EventListener creados para detectar los cambios en la variable estado. Cada vez que el estado de un Lote pasa a ocupado uno de los eventListener agrega el valor de la fecha y del objeto que lo envia a un String y cuando ese mismo objeto cambia a desocupado denuevo, el segundo EventListener agrega la fecha de salida y envia los datos a la clase ManejoDatos para ser guardados en el txt y luego limpia la variable para una nueva ocurrencia del evento.