



Sistemas de Computación

Proyecto Final de Materia

Sistema de bicicletas compartidas

Docentes: Solinas, Miguel Ángel. Jorge, Javier.

Integrantes:

Carranza Agustín	36.947.736
Lenta Luis	37.438.281
Monsierra Lucas	37.488.388
Fissore Lorenzo	19.064.472
Blanco Lucas	39.056.912
Sarquis Tomas	39.884.977

Cohorte 2020

Proyecto asignado:

Sistema de bicicletas compartidas.

Diseñar e implementar un sistema de bicicletas compartidas que mantenga registro en ambos extremos (bicicleta y celular) de las siguientes acciones:

- Apertura y cierre de bloqueo para iniciar/finalizar uso de bicicleta
- Velocidad media del último minuto
- Ubicación
- Tiempo total de uso

La bicicleta contará con un sistema de comunicación bluetooth para comunicarse con la aplicación de celular que habilite el uso de la bicicleta previa autenticación con un código liberado por la central que gestiona el servicio. El control en la bicicleta se debe poder habilitar/deshabilitar en tiempo de ejecución.

Índice general	
Proyecto asignado	2
Índice general	3
Índice de Figuras	4
Índice de Tablas	5
Introducción	6
Marco teórico	6
Requerimientos	8
Descripción del sistema utilizando SYSML	8
Requerimientos	9
Estructura	9
Comportamiento	10
Descripción y diseño de la solución de hardware.	13
Diseño de la solución de software.	15
Casos de prueba del software/hardware.	18
Conclusiones.	20
Bibliografía.	21

2. Índice de Figuras

Figura 5.1 - CPS	6
Figura 7.1 - Diagrama de paquetes	8
Figura 7.2 - Diagrama de requerimientos	9
Figura 7.3 - Diagrama de bloques	10
Figura 7.4 - Diagrama de bloques internos	10
Figura 7.5 - Diagrama de casos de uso	11
Figura 7.6 - Diag. de actividad - Disp. móvil e integrado	13
Figura 8.1 - Tabla comparativa de modelos Raspberry	13
Figura 8.2 - Solenoide	14
Figura 9.1 - Modelo estático	16
Figura 9.2 - Diagrama de secuencia - Login y Emparejamiento	17
Figura 9.3 - Diagrama de secuencia - Consulta de ubicación	18
Figura 9.4 - Diagrama de secuencia - Consulta de datos	18
Figura 10.1 - Diagrama de caso de prueba CP1	19

3. Índice de Tablas

Tabla 10.1 - Planilla de caso de prueba - Visualización de datos

19

4. Introducción

El sistema ciber físico cuyo diseño se detalla en los subsiguientes capítulos del presente informe pretende implementar de manera eficiente la funcionalidad de un sistema de bicicletas compartidas.

La funcionalidad mencionada comprende, el registro y visualización de datos de uso de cada bicicleta en el sistema, la autenticación y bloqueo de seguridad para preservar las bicicletas, y la aplicación móvil para proveer una interfaz de comunicación con el usuario.

5. Marco teórico

Los Cyber Physical Systems (CPS) se refieren a todas las tecnologías que permiten que los recursos computarizados y los objetos físicos interactúen de manera coordinada, confiable y continua para realizar múltiples tareas de donación. Como se ilustra en la siguiente figura, estos CPS pueden combinar hardware y software de tecnologías físicas, biológicas, de ingeniería y de información.

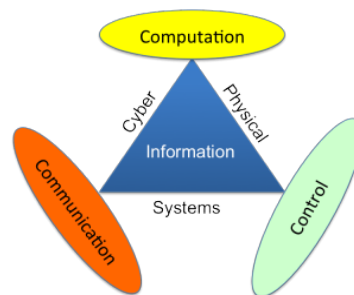


Figura 5.1 - CPS

Los primeros ejemplos de CPS aparecieron en Alemania durante la Segunda Guerra Mundial con sistemas integrados para monitorear las alas de los aviones. Desde entonces, los CPS se desarrollaron principalmente en la industria aeroespacial.

Se podría nombrar el caso de un sistema que controle el nivel de agua de una represa, donde las mediciones tomadas por un sensor harán activar la apertura de las compuertas del dique cuando el agua llegue a una altura determinada.

A diferencia de los dispositivos actuales, la misión principal de los CPS no es la integración en automóviles, juguetes, dispositivos médicos o instrumentos científicos, sino que integra la dinámica de los procesos físicos con los de software y las redes, proporcionando abstracciones y técnicas de análisis, diseño y modelado para todo el integrado.

El potencial económico y social de estos sistemas es increíblemente grande, por lo que se han y están realizando importantes inversiones alrededor de todo el mundo para avanzar en esta tecnología.

Como sabemos, uno de los negocios en auge actualmente en el mundo es el del alquiler de medios de movilidad sustentables, ya sea un monopatín eléctrico o una bicicleta. Basándonos en este crecimiento, creemos que es una excelente oportunidad desarrollar un sistema de bicicletas compartidas, principalmente en la ciudad de Córdoba, donde seríamos pioneros de este negocio.

Esta forma de movilidad se la conoce como micromovilidad o movilidad sustentable. Se utiliza particularmente en zonas urbanas con alta densidad poblacional y su objetivo es brindar una red de medios de transporte ligeros que permitan movernos trayectos cortos. Surge de la necesidad de las personas de desplazarse distancias que son “largas” para hacerlas caminando pero “cortas” para hacerlas con un automóvil. Las principales ventajas que presenta en comparación con los automóviles son:

- Mayor efectividad para tramos cortos.
- Menor contaminación.
- Menor gasto energético.
- Son fáciles de utilizar.
- Mayor libertad para alcanzar el destino y evitar atascos.

Actualmente en Córdoba casi no existen opciones de micromovilidad compartida. Hasta el día de hoy no hay ninguna empresa que brinde el servicio de alquiler de monopatines eléctricos para uso compartido. El mercado del alquiler de bicicletas está centralizado en locales comerciales, la mayoría de venta de bicicletas y se ofrece como un servicio a turistas. Se puede observar una mecánica tradicional con formularios en papel, contacto telefónico y sin control en tiempo real de la bicicleta. La plataforma Bewee¹, que se encuentra en fase experimental y es la única que disponible en Córdoba actualmente, da un paso más y concentra en un espacio común personas que quieren alquilar y dueños de bicicletas. Aún así, no se tiene datos del vehículo ni estadísticas. Además, cada vez que un usuario termina de utilizar la bicicleta, debe desplazarse hasta uno de estos espacios comunes para dejarla, lo que se vuelve incómodo y tedioso.

El sistema que proponemos está compuesto por un grupo de bicicletas distribuidas estratégicamente en los puntos de mayor concurrencia de los barrios más poblados, como por ejemplo, el Paseo del Buen Pastor, la Plaza San Martín, la terminal de ómnibus, etc. Si un usuario desea utilizar una de nuestras bicicletas, lo único que debe hacer es registrarse a través de nuestra aplicación móvil, desbloquearla con su celular y comenzar a disfrutarla. En el momento que el usuario decide terminar de usarla, basta con dejar la bicicleta en cualquiera de la zonas habilitadas y registrarlo en su celular.

Cada una de ellas, dispone de un dispositivo de control encargado de realizar la gestión, registro de datos, desbloqueo y bloqueo de la bicicleta mientras no está siendo

¹https://www.cba24n.com.ar/tecnologia/bewee--una-opcion-para-mejorar-el-traffic-en-la-ciudad_a5df24f3f5b75f36f25a5a35d

utilizada. Los datos recopilados, van siendo actualizados en la aplicación, así el usuario puede conocer la velocidad con la que se desplaza, el tiempo que lleva utilizando la bicicleta y la posición en la que se encuentra. Estas características se vuelven el principal diferenciador frente a la competencia, ya que actualmente ellos solo permiten el bloqueo y desbloqueo de la bicicleta mediante el celular pero no muestran ningún tipo de información.

6. Requerimientos

A continuación se detallan los requerimientos funcionales:

- **Registro de Datos:** El dispositivo integrado debe registrar la velocidad, ubicación y tiempo de uso.
- **Sistema de Bloqueo:** El dispositivo integrado debe permitir bloquear y desbloquear la bicicleta.
- **Comunicación Bluetooth:** El dispositivo integrado en la bicicleta debe permitir vincularnos vía Bluetooth con nuestro celular.
- **Aplicación Móvil:** El sistema debe disponer de una aplicación encargada de la interacción con el sistema integrado en la bicicleta.
- **Autenticación:** La aplicación debe permitir el acceso a usuarios registrados previamente.
- **Control de Sistema de Bloqueo:** La aplicación debe permitir bloquear y desbloquear la bicicleta a usuarios autorizados.
- **Visualización de Datos:** La aplicación debe mostrar al usuario los datos registrados por el usuario.

7. Descripción del sistema utilizando SYSML

Para estructurar y describir nuestro proyecto utilizamos un diagrama de paquetes, en el que se definen tres grupos bien diferenciados, Comportamiento, Estructura y Requerimientos. Dentro de cada uno de estos grupos se incluyen los correspondientes diagramas que ayudan a modelar el diseño de nuestro sistema.

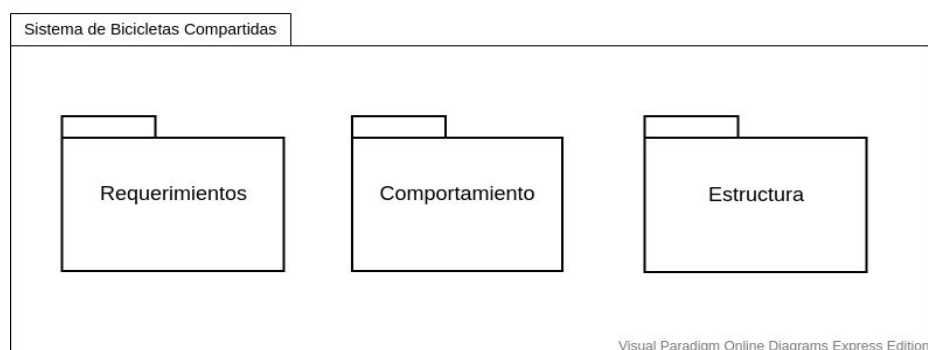


Figura 7.1 - Diagrama de paquetes

Requerimientos

Para la descripción de requerimientos funcionales de nuestro sistema se utiliza un único diagrama, donde vemos cada uno de ellos representados por bloques y cómo se relacionan. En la Figura 7.2 encontramos dicho diagrama. Para una mayor organización dividimos los requerimientos funcionales, descritos arriba, en dos principales que a su vez se descomponen en otros más particulares.

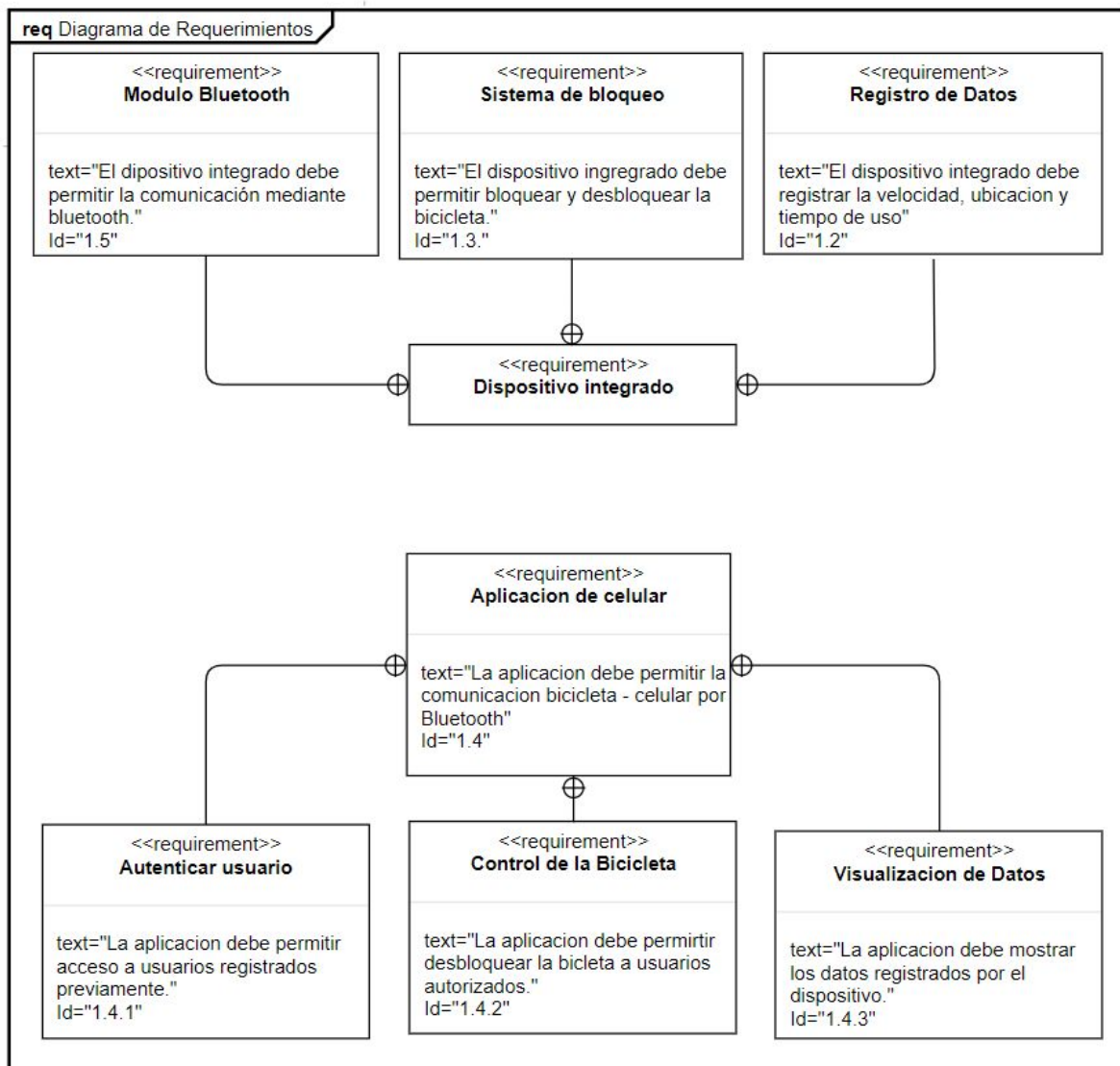


Figura 7.2 - Diagrama de requerimientos

Estructura

En la Figura 7.3 visualizamos un diagrama de bloques. En él creamos un bloque raíz denominado *contexto* a partir del cual se relacionan todos los elementos necesarios.

El contexto está compuesto por bicicletas que pueden ser utilizadas por los usuarios. Un dispositivo integrado en la bicicleta que a su vez cuenta con un módulo

GPS, un sistema de bloque y un módulo de Bluetooth. Por último un celular que cuenta con un módulo Bluetooth para la comunicación del usuario con el dispositivo integrado y uno o varios sistemas de bases de datos que el celular utiliza para la autenticación.

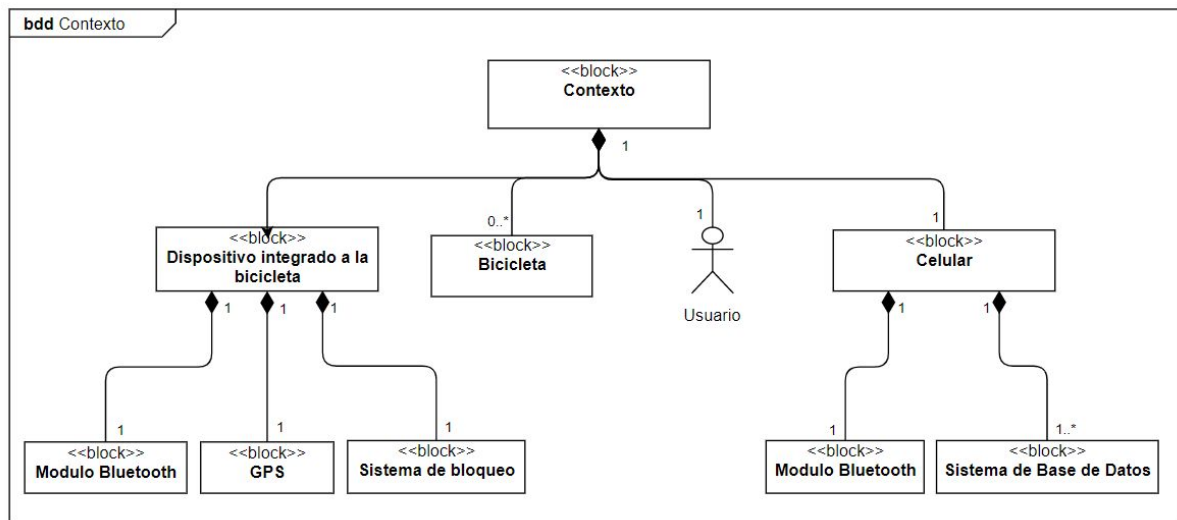


Figura 7.3 - Diagrama de bloques

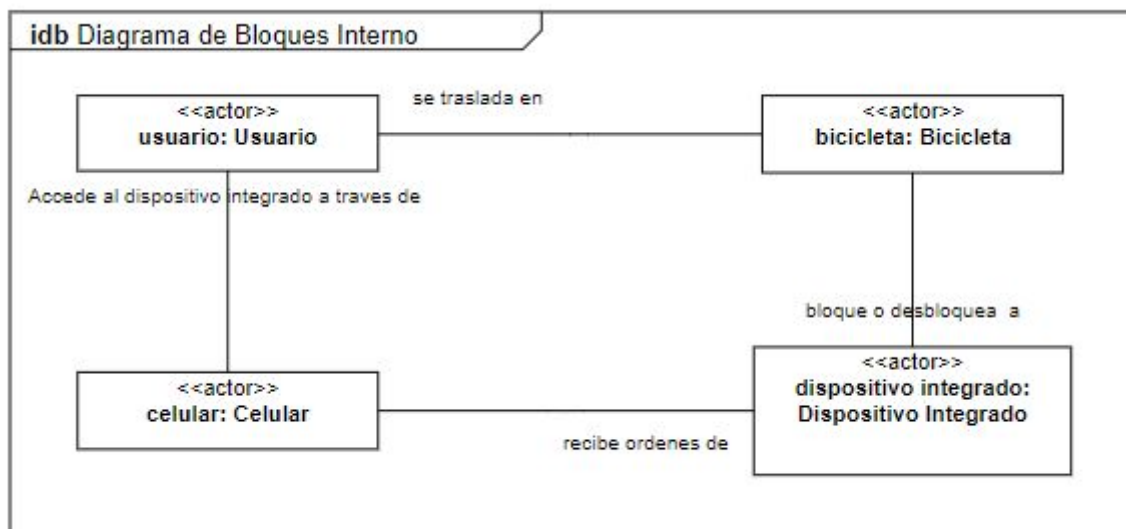


Figura 7.4 - Diagrama de bloques internos

Comportamiento

Para observar y comprender el funcionamiento de nuestro sistema, utilizamos un diagrama de casos de uso (Figura 7.5). En éste podemos ver los principales servicios que brinda nuestro sistema y como el usuario interactúa con ellos. En el diagrama podemos distinguir tres subsistemas, la aplicación móvil, la bicicleta y el dispositivo integrado.

Para comenzar con el uso de la bicicleta, el usuario debe autenticarse a través de la aplicación móvil. Una vez autenticado se puede conectar al dispositivo integrado en la bicicleta mediante Bluetooth y desbloquearla. Durante el uso de la bicicleta, el usuario

puede visualizar diferentes datos, que recopila el dispositivo integrado, en la aplicación de su celular. Una vez que termina de utilizar la bicicleta, puede dejarla donde quiera y bloquearla.

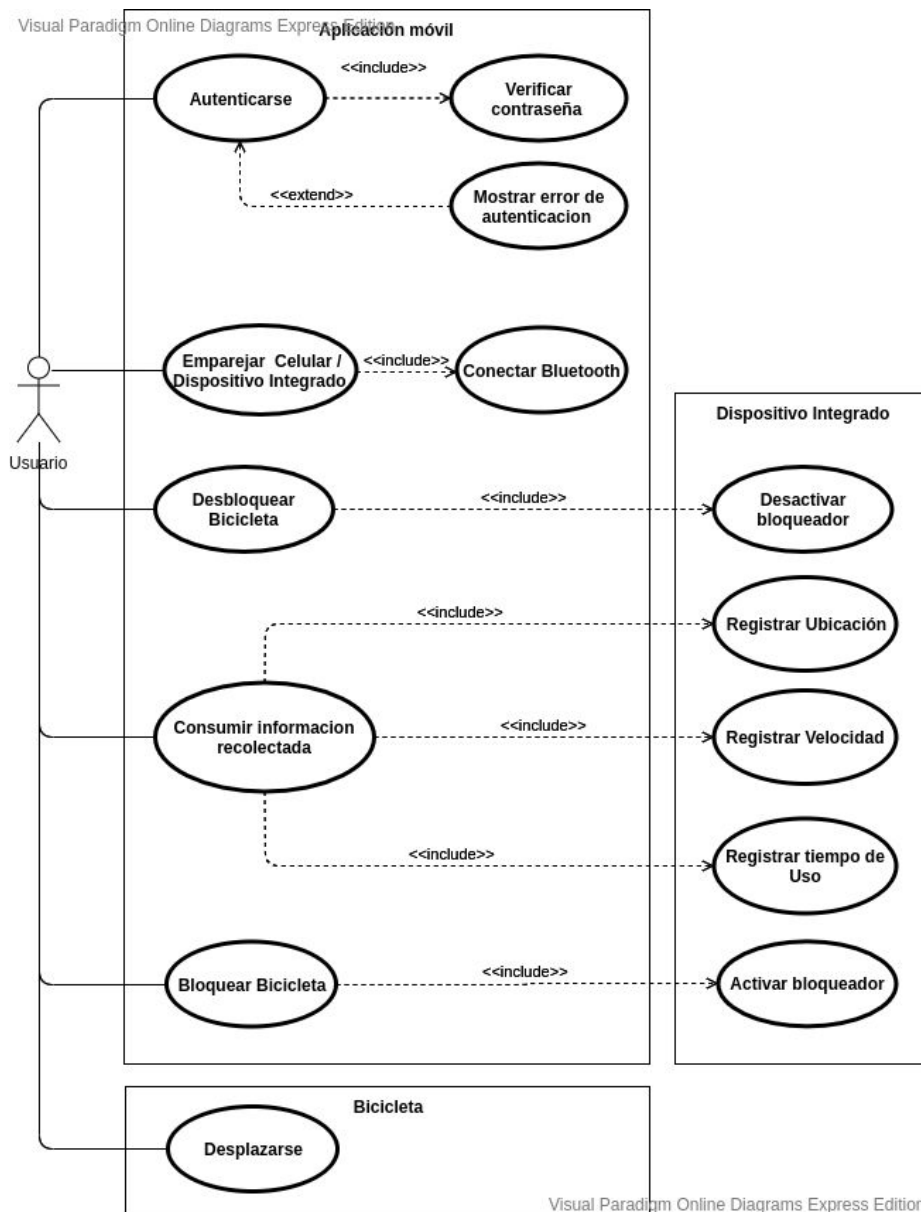


Figura 7.5 - Diagrama de casos de uso

Finalmente, en la Figura 7.6 se representa a través de una diagrama de actividades la interacción entre el dispositivo móvil y el sistema integrado en la bicicleta.

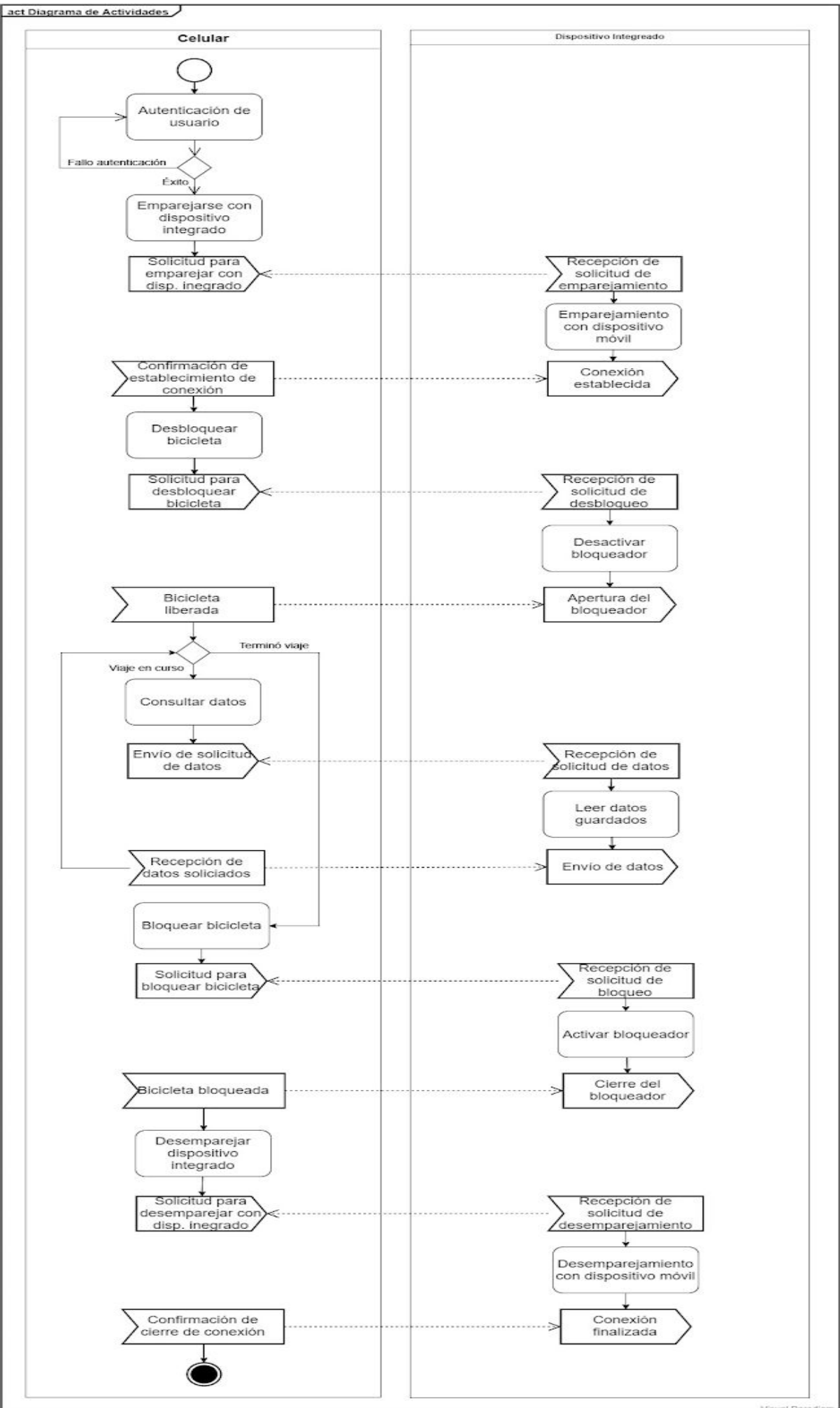


Figura 7.6 - Diag. de actividad - Disp. móvil e integrado

8. Descripción y diseño de la solución de hardware.

Determinados ya los componentes del sistema y la relación que existe entre ellos en las secciones previas, basamos nuestras decisiones de diseño y elección de componentes de hardware para materializarlos.

La parte clave de nuestro sistema, el dispositivo integrado está conformado por un microordenador Raspberry Pi unido a una serie de sensores necesarios para la recopilación de datos, entre los cuales se tiene un sensor GPS, un dispositivo Bluetooth y un mecanismo de apertura y cierre.

El microordenador elegido es el RASPBERRY PI 3 MODEL B. Es una single board computer (SBC) de bajo costo para propósitos educativos. Este modelo destaca de sus anteriores en que incluye de manera integrada un chip para WiFi y Bluetooth. Sus características son:

	Raspberry Pi 3 Model B	Raspberry Pi Zero
Introduction Date	2/29/2016	11/25/2015
SoC	BCM2837	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz
Instruction set	ARMv8-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM
Storage	micro-SD	micro-SD
Ethernet	10/100	none
Wireless	802.11n / Bluetooth 4.0	none
Video Output	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI
GPIO	40	40
Price	\$35	\$5

Figura 8.1 - Tabla comparativa de modelos Raspberry

Al tratarse de un prototipo se decide implementar el sistema en el modelo Pi 3 Model B a modo didáctico y educativo. Puede ahorrarse un monto considerable de dinero si se opta por un modelo más económico de placas Raspberry Pi, como por ejemplo la Raspberry Pi Zero con un costo de 5 dólares, u optar por otro tipo de hardware más barato, como por ejemplo Arduino, siendo posible conseguir placas de desarrollo a menos de 5 dólares, pero lo que no se paga en precio, se paga en rendimiento, ya que poseen menor potencia de la familia Raspberry Pi. Si se piensa en producción a gran escala, se debe migrar a un sistema con un micro más barato.

El Sistema de Posicionamiento Global (GPS, Global Positioning System) es un sistema que permite determinar en toda la Tierra la posición de cualquier objeto con una

precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de Estados Unidos, y actualmente es propiedad de la Fuerza Espacial de los Estados Unidos.

Para determinar su posición, un usuario utiliza cuatro o más satélites y utiliza la trilateración. Existen actualmente en el mercado diversos módulos de distintos precios y calidades que implementan esta funcionalidad. Los más conocidos y populares son aquellos basados en la familia de microcontroladores NEO-6 de u-blox, el cual utilizaremos en el presente. Estos normalmente implementan una comunicación UART TTL. Para comunicar este módulo con Raspberry y mantener niveles de señal correctos podemos usar un módulo *level shifter*.

Para el sistema de cierre y bloqueo hay disponibles algunas cerraduras eléctricas basadas en solenoides (electroimanes). Esto conectado a un relay compatible con la computadora central, permite enviar la señal de activación a la traba y liberar o bloquear la bicicleta.



Figura 8.2 - Solenoide

Al tratarse de un prototipo, el sistema de cierre y bloqueo en esta instancia no es de vital importancia, siendo posible implementarla con sistema básico y simple como el descrito anteriormente, ya que, a diferencia del GPS por ejemplo, no transmite información, sino que únicamente recibe señales por las que reacciona. En el caso de un modelo de producción, se debería diseñar un sistema más robusto y resistente para que la bicicleta no sea utilizada sin permiso previo, situación que puede darse en este prototipo con simplemente violentar este tipo de cerradura.

9. Diseño de la solución de software.

En el apartado anterior se definieron los componentes de hardware que mapean cada uno de los requerimientos de nuestro sistema. Partiendo de esta base, se continúa con la descripción del modelo dinámico y estático del software que corre sobre los distintos componentes de hardware.

Para describir la parte estática del sistema se utiliza un diagrama de clases, el cual nos permite apreciar aspectos relacionados a la organización del software, mostrando las clases del sistema, sus atributos, métodos y las relaciones entre los objetos. En la Figura 9.1 se presenta este diagrama, el cual contiene 2 paquetes, uno correspondiente a la aplicación móvil y otro al dispositivo integrado.

En el primer paquete nombrado, se pueden apreciar todas las clases relacionadas con la aplicación móvil, donde se encuentra la clase Login, encargada de toda la gestión de autenticación del usuario y se relaciona con la clase Menú que es con la que va a interactuar en mayor medida el usuario y nos da acceso a la mayor parte de las funcionalidades que se describieron en los requerimientos, como por ejemplo mostrar datos relacionados a velocidades y tiempo encuadrados en la clase Datos, ubicación, contenidos en la clase Ubicación o emparejamiento con el dispositivo integrado en la clase ModuloBT.

Estas últimas 4 clases, Menú, Datos, Ubicación y ModuloBT utilizan y están en constante comunicación con la clase Dispositivo en el segundo paquete realizando las correspondientes peticiones que se describirán en las Figuras 9.2., 9.3 y 9.4, como por ejemplo emparejar con dispositivo integrado, consultar datos o ubicación.

En cuanto al segundo paquete, se describen 2 clases, una Dispositivo, ya mencionada, que es la encargada de gestionar y recabar todos los datos que generan los módulos implementados y otra Driver, la cual se implementa para llevar a cabo las funcionalidades de apertura y cierre del bloqueador.

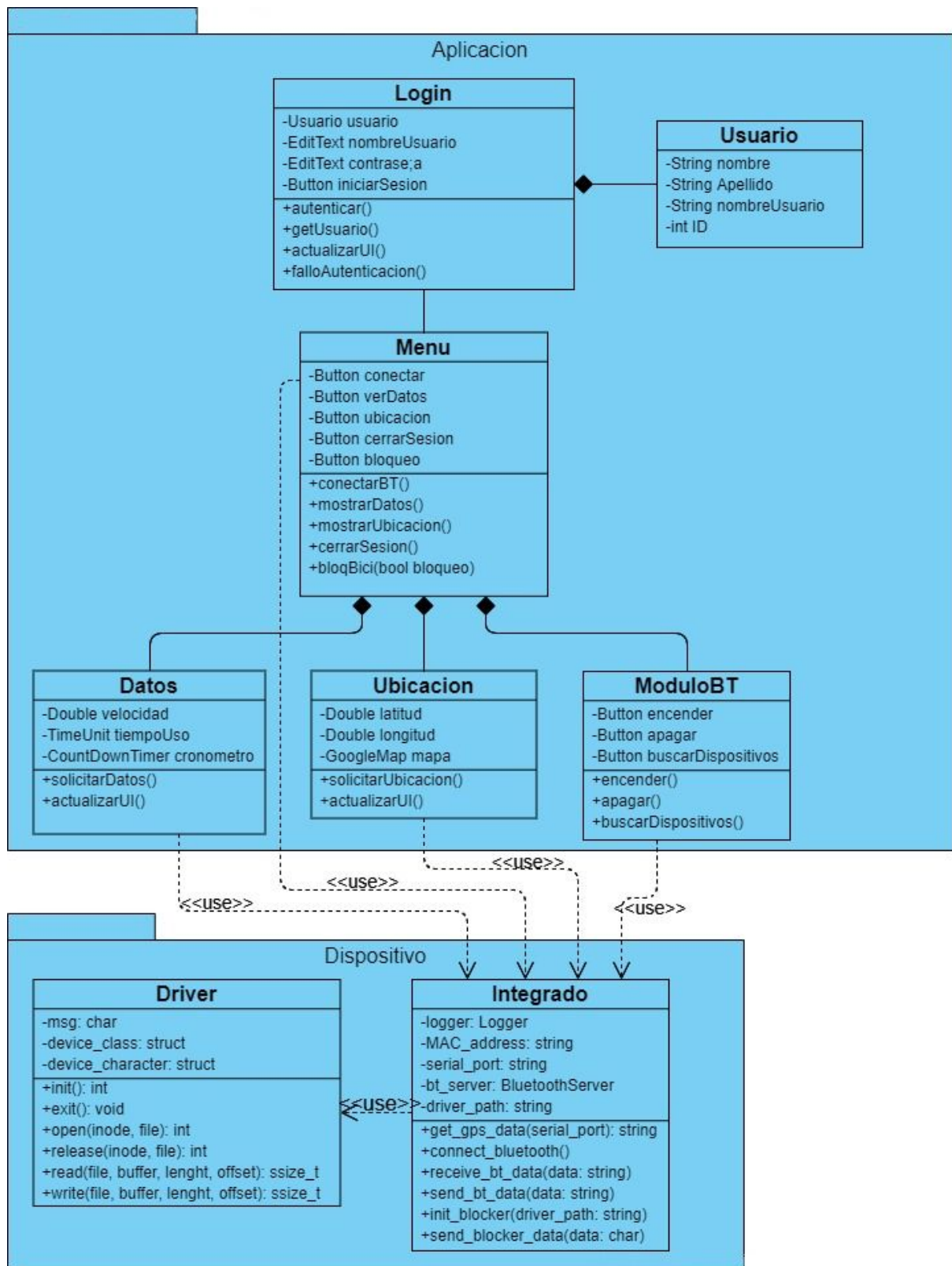


Figura 9.1 - Modelo estático

El modelo dinámico del software queda descrito por los diagramas de secuencia de las Figuras 9.2, 9.3 y 9.4. Decidimos dividirlo en 3 partes para mostrar en detalle la secuencia vertical de mensajes que pasan entre los elementos en una interacción y que la figura no sea extremadamente extensa.

En la Figura 9.2 se aprecia, en primera instancia, el intercambio de mensajes en entre el usuario en una autenticación exitosa. Una vez resuelto el login, el usuario decide emparejarse disparando una secuencias de acciones donde se llevan a cabo intercambios de mensajes entre el dispositivo móvil y el integrado que está escuchando constantemente a la espera de nuevas conexiones. Finalizado el emparejamiento, se devuelve el control a usuario para que pueda realizar nuevas peticiones.

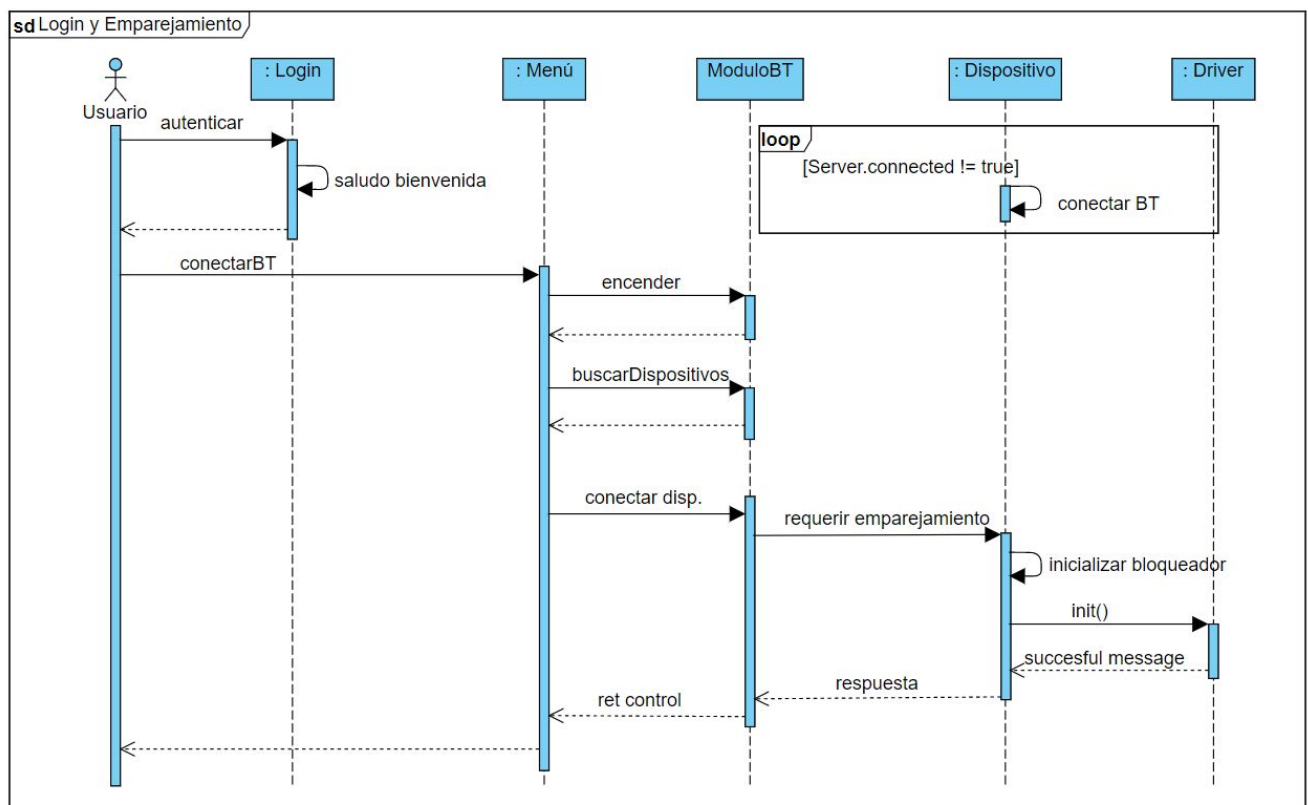


Figura 9.2 - Diagrama de secuencia - Login y Emparejamiento

Partiendo de la premisa anterior, en las Figuras 9.3 y 9.4, se describe el intercambio de mensajes que sucede en caso de que el usuario solicite consultar la ubicación o los datos acerca de velocidad y tiempo de uso respectivamente.

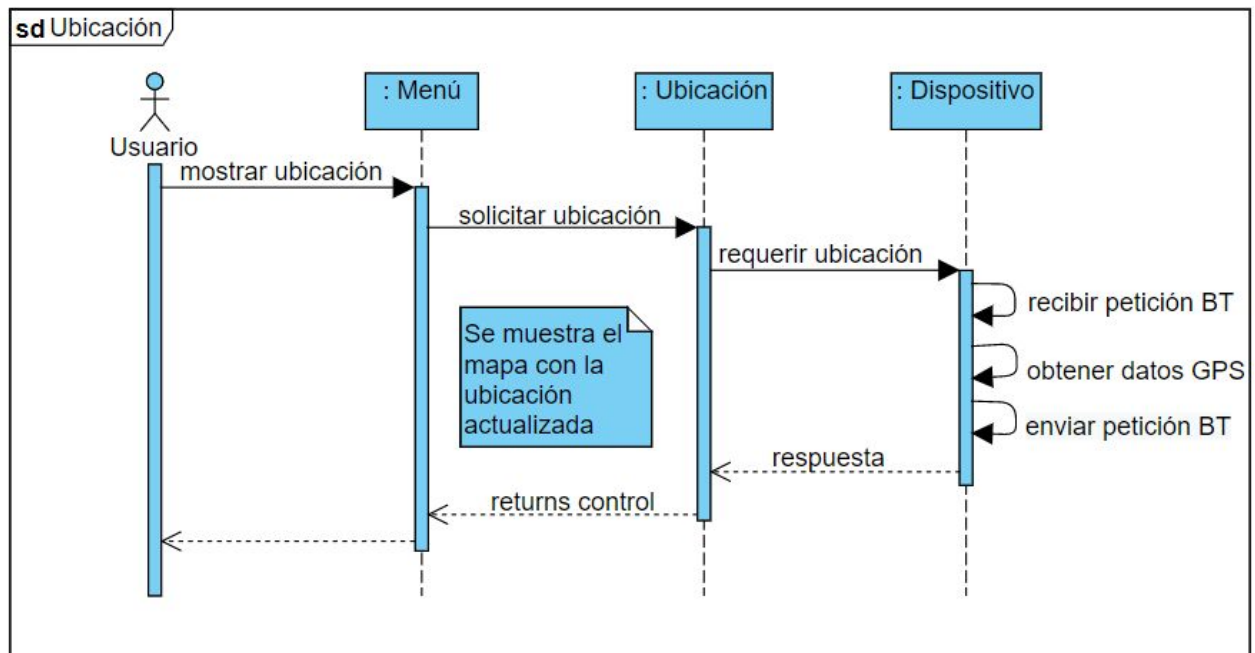


Figura 9.3 - Diagrama de secuencia - Consulta de ubicación

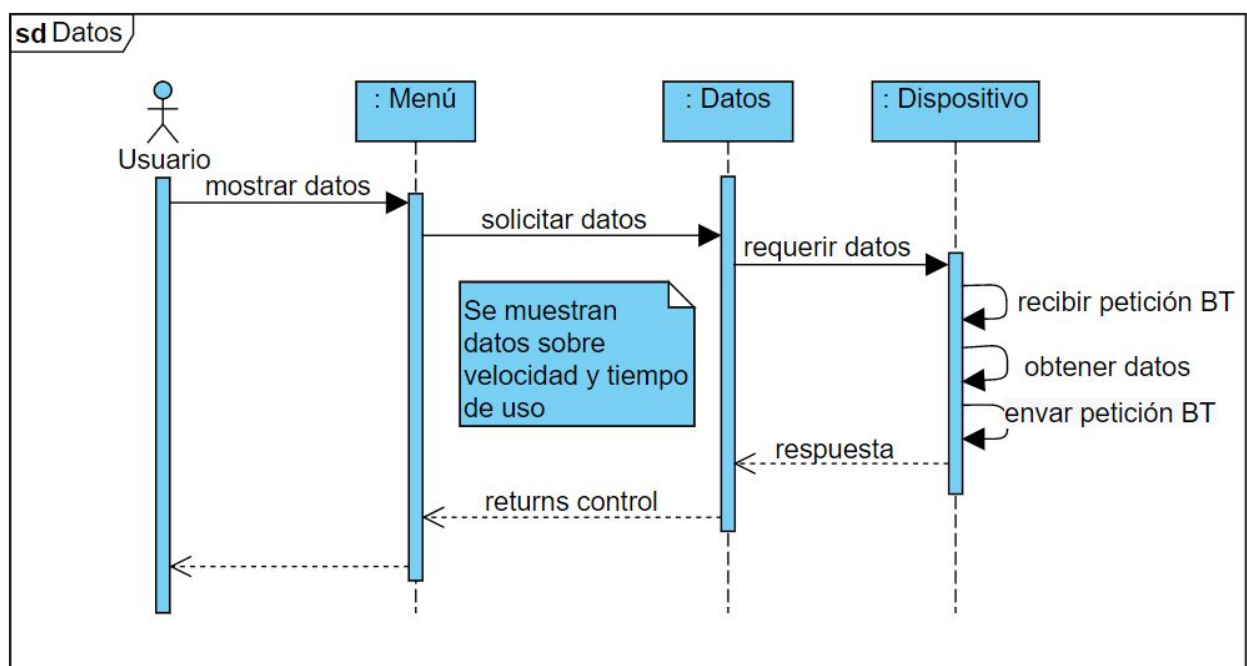


Figura 9.4 - Diagrama de secuencia - Consulta de datos

10. Casos de prueba del software/hardware.

El requisito funcional abordado es el de **visualización de datos**:

ID	CP1
Descripción	Visualización de los datos registrados
Acción	Pulsado de botón "Ver datos"
Resultado esperado	Visualización de los siguientes datos: <ul style="list-style-type: none">• Velocidad media• Tiempo de uso
Pre-condiciones	El usuario se encuentra logueado correctamente.
Entradas	-
Post-condiciones	-

Tabla 10.1 - Planilla de caso de prueba - Visualización de datos

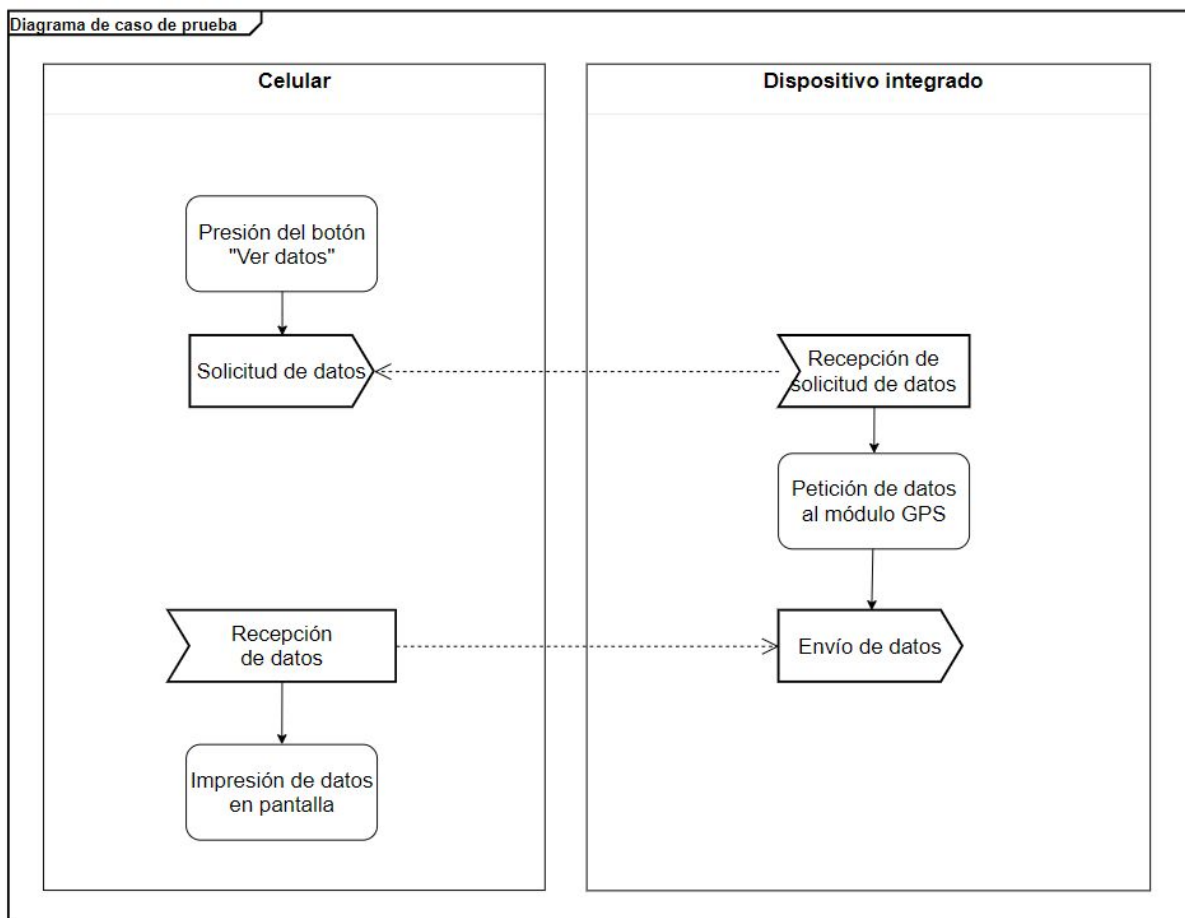


Figura 10.1 - Diagrama de caso de prueba CP1

11. Conclusiones.

A partir del proyecto asignado comenzamos un estudio detallado de los distintos proyectos que existen relacionados al nuestro en el mundo. Ver su funcionamiento, aplicaciones, modelos, entre otros, nos sirvió para poder comenzar a pensar una base y definir los requisitos del nuestro.

Redactarlos nos precisó tarea de resumen, y de pensar claramente cuáles eran los requisitos mínimos que necesitábamos para que nuestro sistema funcione.

Con esto en claro, utilizando los diagramas que proporciona SYSML iniciamos el diseño de nuestro sistema. En nuestro caso utilizamos la plataforma online Visual Paradigm² con la cual pudimos trabajar teniendo los distintos diagramas en un repositorio compartido por todos los miembros del grupo, facilitando la distribución de tareas.

Uno de los inconvenientes con el cuál nos cruzamos al realizar los diagramas fue el poder abstraerse de pensar en el código y la implementación física de nuestro sistema en dispositivos puntuales. Aprendimos a diseñar pensando en general, a más alto nivel.

La parte clave de la implementación, como ya se ha mencionado, es la relacionada al microordenador Raspberry Pi y su comunicación con los sensores necesarios para la recopilación de datos.

En la etapa de prototipado, se han tomado ciertas concesiones, por ejemplo, las relacionadas a sistema de cierre. En el caso de un modelo de producción, se debería implementar un sistema más robusto, lo cual, gracias a las características del diseño, solo implicaría el uso de componentes con características acordes a los requerimientos específicos de la producción. Incluso si ciertas interfaces entre los componentes se vieran modificadas el diseño seguiría manteniendo su estructura y se adaptaría fácilmente a los cambios.

² "Visual Paradigm." <https://www.visual-paradigm.com/>.

12. Bibliografía.

SysML by Example Tutorial: Griffin Space Vehicle Project. (n.d.). SysML.org. Consultado 27 Mayo, 2020, desde <https://sysml.org/res/sysml-example-tutorial/index.html>

pmoinformatica.com. (n.d.). *Requerimientos funcionales: Ejemplos*. Consultado 27 Mayo, 2020, desde <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>

Cyber Physical Systems (CPS) 2B1stconsulting. (n.d.). Consultado 27 Mayo, 2020, desde <https://www.2b1stconsulting.com/cyber-physical-systems-cps/>