Proyecto de Sistemas de Computación
Sistema de Seguridad de Caja Fuerte

Grupo 2

Cohorte 2020

Docentes:
Jorge, Javier
Solinas, Miguel Angel

Alumnos:
Argayo, Juan Segundo
Bosack, Facundo Matias
Navarro, Sebastian
Pichetti, Augusto Gabriel
Piñero, Tomas Santiago
Salse, Lucas Antonio

Índice

Índice de figuras

Índice de tablas

1. Introducción

Hoy en día, la necesidad de monitorear y conocer el estado de una caja fuerte sin tener que encontrarse físicamente en el mismo lugar está siendo cada vez mayor. Analizando datos estadísticos de distintos países se ve que en Estados Unidos, 1 de cada 13 habitantes alquila una caja fuerte ya sea en empresas privadas o en un banco. En Francia, lo hacen 1 de cada 5 habitantes, en Suecia 1 de cada 3 y en Argentina 1 de cada 55 habitantes alquila una de ellas [2].

Es importante mencionar que, en la actualidad, la mayoría de los dispositivos están conectados a Internet, es decir que, se necesitan diseños guiados por un concepto de red que implique la interacción con otros elementos del mundo físico a través de entradas y salidas.

Es por eso que el objetivo de este trabajo es una solución innovadora que resuelva esta necesidad. Se realizará el diseño y la implementación de un sistema de seguridad que detecte las acciones de apertura y cierre de la caja. Así, los dueños de las mismas podrían ver en tiempo real el estado de su caja fuerte.

2. Marco teórico

2.1. *Cyber-Physical System*

El presente proyecto describe el diseño e implementación de un prototipo de sistema de seguridad de caja fuerte. El sistema está compuesto por *hardware*, software, redes y procesos físicos. Así definido es posible abordar su estudio como *Cyber-Physical System* (CPS), un concepto relativamente nuevo que ayudará a enfrentar su análisis, diseño e implementación.

Como puede verse en la Fig. **1**, en un CPS, las computadoras y redes integradas se encargan de monitorear y controlar ciertos procesos físicos en tiempo real. En este caso en particular son los eventos y acciones que se producen sobre la caja fuerte, los cuales generan una retroalimentación al usuario permitiéndole disponer de información útil para tomar decisiones [1].

A diferencia de los sistemas embebidos tradicionales, que consisten en dispositivos independientes y aislados, un CPS es diseñado bajo un concepto de red que interactúa con otros elementos del mundo físico a través de salidas y entradas. Se destaca por sus capacidades, adaptabilidad, escalabilidad, resiliencia, seguridad y usabilidad que superan notablemente al de un sistema embebido (ver Fig. **2**).

Por tratarse de un prototipo, se busca cumplir estrictamente con un conjunto básico de requerimientos funcionales y luego ampliarlos, rediseñarlos o escalarlos. Es decir, que sea capaz de tener nuevas funcionalidades sin que esto genere un esfuerzo mayor en su diseño e implementación.

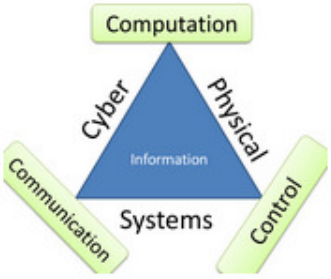

Figura 1: CPS.

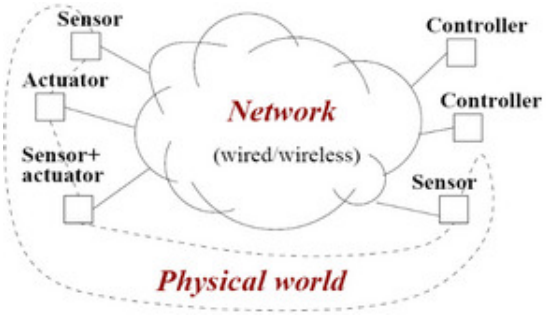

Figura 2: Ejemplo CPS.

2.2. Productos similares en el mercado

En el mercado se pueden encontrar sistemas similares, alguno de ellos son:

2.2.1. *SafeWizard Multi Door Access Control Safe Lock*

Características:

- Tiempo de retardo para cualquier puerta de 1-99 minutos y periodo de apertura de 1-99 minutos.
- Programable tiempo de bloqueo para cualquier puerta.
- Penalización, Bloqueo por códigos de usuario incorrectos (programable).
- Alarma silenciosa de atraco en el último dígito del PIN.
- Software de fácil navegación. Con menús que son intuitivos y fáciles de entender.
- Capacidad de diagnósticos del sistema con modos de prueba.
- Dispositivo de comunicación es totalmente segura con el sistema de comunicaciones de la red de área local VaultLAN® RS485.

2.2.2. *Tech MASTER*

Fabricante: *Tecnosicurezza locks and security systems*[3].

Para determinar un precio sobre algunos de sus productos es necesario contactar con la empresa.

Italia: `info@tecnosicurezza.it`

EEUU: `info@usatecno.com`

España: `info@tecnosicurezza.es`

Características:

- 90 usuarios
- Retardo de apertura 1-99 min.
- Penalización por intentos erróneos.
- Ventana de apertura 1-19 min.
- Retardo de tiempo alternativo.
- Agrupación de usuarios.
- Bloqueo semanal.
- Función bloqueo/desbloqueo.
- Periodos de apertura y cierre.
- Test de sistema.
- Alarma silenciosa y bloqueo remoto (opcional).
- Programación por teclado.
- Conectividad IP Con administración y monitoreo total del sistema (opcional).
- Memoria de eventos (900 eventos con fecha y hora).
- Software de gestión y monitoreo IP.

2.2.3. *LA GARD – ADITGARD*

Características:

- Cerradura multi-usuario que provee control de acceso superior.
- Programación desde software.
- Capacidad para 8 usuarios y 1 *Manager*.
- Código *manager* que gestiona hasta 9 usuarios (Añadir/Eliminar/Habilitar/Deshabilitar).
- Penalización de la cerradura por varios intentos de combinación erróneos 512 eventos de auditoría con hora y fecha. (requiere software)
- Tiempo de retardo de 1 a 99 minutos.
- Señalización de batería baja.
- Alimentación a través de la botonera mediante una batería alcalina de 9V.
- Batería de larga duración.
- Software de auditoría.

Algunos de estos productos similares que se pueden encontrar en el mercado son realizados por empresas dedicadas directamente con la seguridad de cajas fuertes a niveles altos, como ser orientado a bancos o correos. Algunas de estas empresas son las siguientes:

LIZ SAFE[4]

Una compañía especialista en la fabricación y desarrollo de todo tipo de equipos de seguridad en la línea de bancos, comercial, hotelera y hogar que cumple con altos estándares en temas de diseño, calidad y tecnología. Dentro de sus productos se encuentran cajas fuertes blindadas, contra incendio, contra robo, consignatarias, para armas, tipo hotel, bóvedas, pasa documentos, cerraduras, ventanas y puertas blindadas.

Contacto: `cajasfuertes@lizsafe.com`

AMSECUSA[5]

American Security Products Company (AMSEC) es el proveedor más conocido del mundo de cajas de seguridad y soluciones de seguridad. Algunos precios de productos relacionados son proporcionados por *GunSafes*, en el sitio web de la empresa pueden verse y completarse el formulario para contactarse para más detalles.

3. Requerimientos

Los requerimientos funcionales del sistema son los siguientes:

- El sistema debe detectar la acción de cierre y apertura de una caja fuerte.
- El sistema debe registrar inmediatamente la ocurrencia del evento y notificar a un celular registrado.
- El control de detección debe poder habilitarse y deshabilitarse en tiempo de ejecución desde un celular registrado y de forma local.
- Se debe poder registrar un celular para recibir notificaciones, habilitar y deshabilitar el control de forma remota.
- El sistema deberá actualizar al arranque, su fecha y hora, desde un servidor de NTP (`https://www.pool.ntp.org/zone/es`).
- El sistema debe contar con un registro de los siguientes eventos:
  - **Tipo de acción**: apertura, cierre, deshabilitación del sistema, envío de notificación al celular y recepción de la lectura de la notificación enviada.
  - ***Timestamp***: guardar en el registro el valor de la fecha y hora en la que se realizó la acción.

4. Diseño a nivel sistema utilizando *SysML*

El diagrama de paquetes del sistema de la caja fuerte realizado con *SysML*[6] es el siguiente:

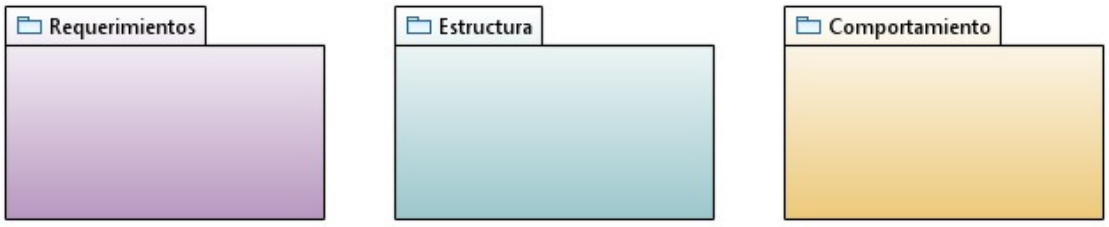

Figura 3: Paquetes *SysML*.

Las Fig. **4, 5 y 6** muestran los diagramas de requerimientos, estructura y comportamiento, respectivamente.

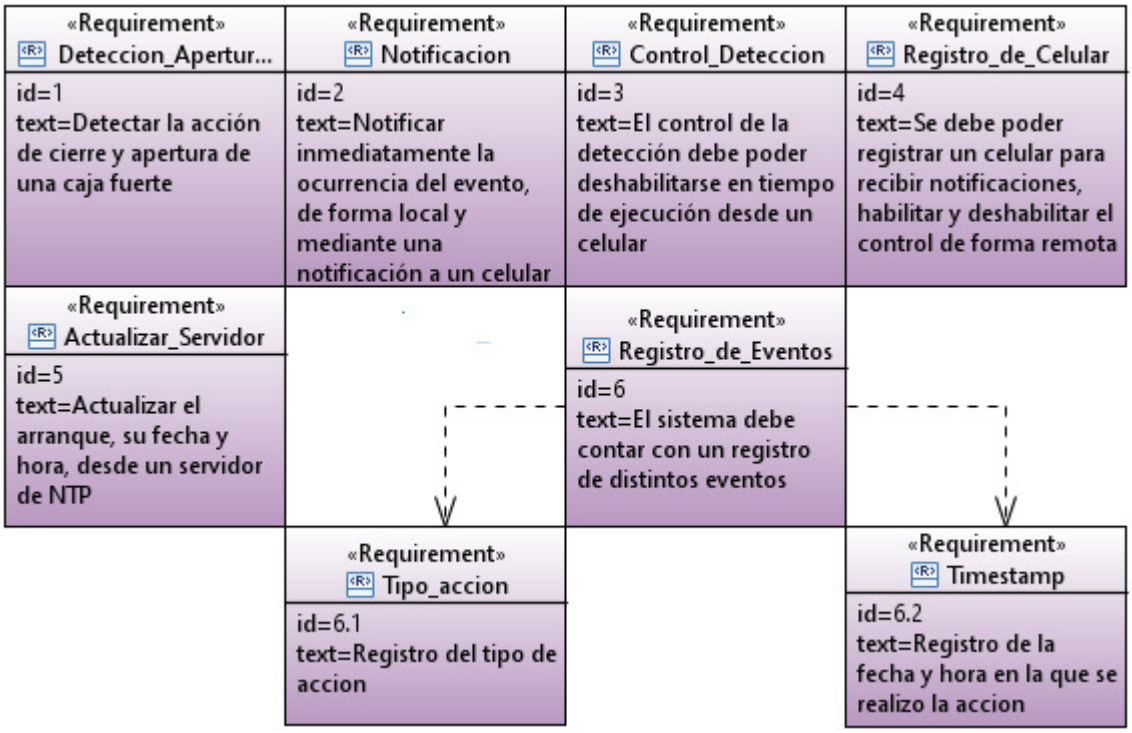

Figura 4: Diagrama de requerimientos.

En la estructura del sistema, tenemos el siguiente diagrama de contexto:

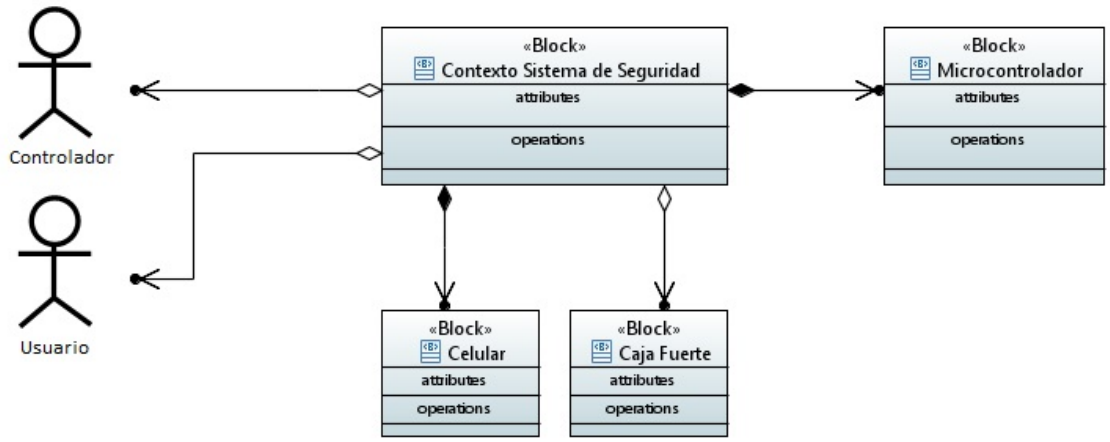

Figura 5: Diagrama de contexto.

Por último, se muestra el diagrama de bloques interno del contexto del Sistema de Seguridad

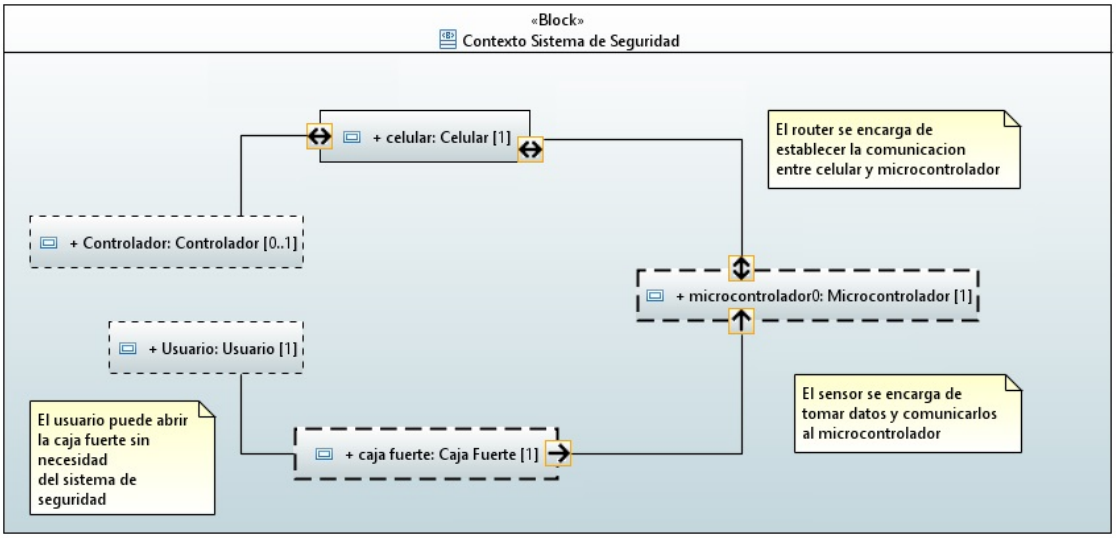

Figura 6: Diagrama de bloques interno.

El sistema inicia con un actor interactuando con la caja fuerte. La acción detectada se envía al sistema de seguridad, que se a su vez la reenvía al celular, quien puede activar y desactivar el sistema de seguridad remotamente.

Cada vez que se detecta uno de los eventos de interacción, se guardan en un registro tanto en el microcontrolador como en el celular.

5. Modelo de comportamiento del sistema

En la Fig. **7** se identifican las diversas formas de utilizar el sistema, tanto desde el punto de vista de un Usuario que abre y/o cierra la caja fuerte, como desde un Controlador (usuario autorizado para el sistema) quien puede activar, desactivar y/o leer los *logs*.

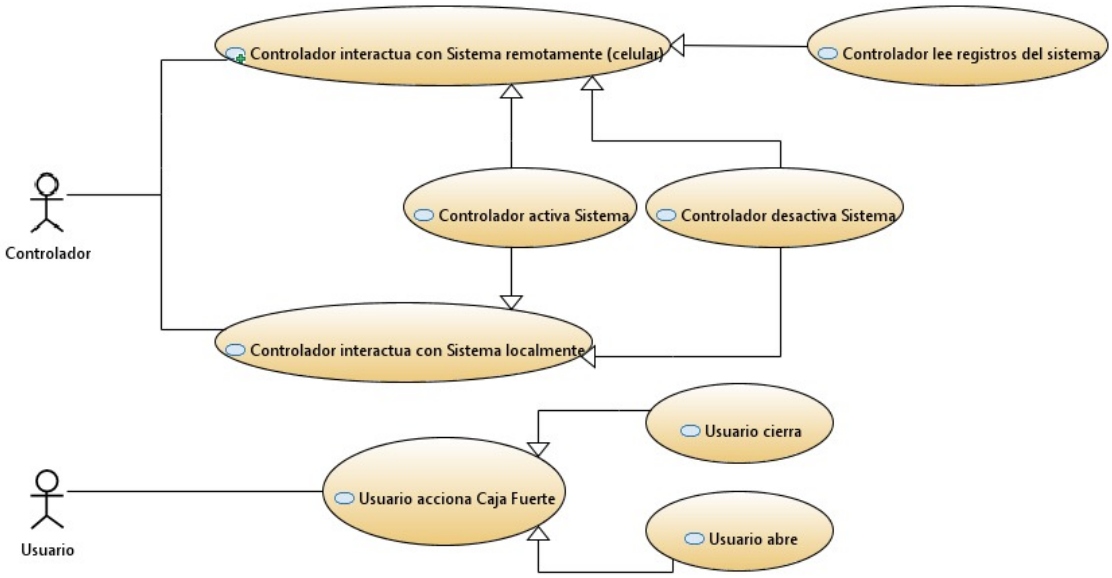

Figura 7: Diagrama de casos de uso.

6. Distribución de requerimientos funcionales en componentes del sistema

Para poder determinar la distribución de los requerimientos funcionales en cada uno de los componentes del sistema se realizó una tabla en donde se puede ver claramente qué requerimiento hace uso de qué componente.

En la Tabla **1** se marcan los requerimientos que se implementarán con el símbolo '✓' y los que no con 'X'.

	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5	Req. 6
microcontrolador	✓	✓	✓	✓	X	✓
sensor	✓					
celular		✓	✓	✓		X

Tabla 1: Matriz de trazabilidad de los requerimientos.

7. Descripción y diseño de la solución de *hardware*

El microcontrolador que va a ser utilizado como centro del sistema es una *Raspberry PI 3 Model B+*[7]. Este microcontrolador cuenta con las siguientes características, más que suficientes para nuestros propósitos:

- CPU Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- Memoria principal 1GB LPDDR2 SDRAM
- Conectividad WiFI 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac wireless LAN
- Bluetooth 4.2
- Gigabit Ethernet por USB 2.0 (300 Mbps)
- Soporte de Power-over-Ethernet (PoE)
- Fuente de energía 5V/2.5A DC.

La elección de este microcontrolador nace de las amplias posibilidades que nos brinda a la hora de diseñar nuestro sistema y programar el mismo. Sumado a esto, entre los integrantes del grupo se cuenta con dicho componente, por lo que nos evita tener que incurrir en el costo de uno nuevo.

Con respecto al sensor utilizado, se decidió utilizar dos simples cables en cuyos extremos contarán con un material conductor. Ambos se conectarán a la *Raspberry Pi* y se indicará la apertura de la puerta cuando los extremos conductores se aparten, provocando un cambio de nivel de tensión y por lo tanto un correspondiente cambio de estado en una *flag* interna dentro de nuestro software. Análogamente, se detectara el cerrado de la puerta cuando los cables entren en contacto.

Por último, el celular del usuario que interactuará con el sistema de seguridad, corre un sistema operativo (SO) *Android*[1] 7 o superior. En principio el sistema solo correrá sobre este SO por ser el más adoptado en el mercado.

El celular *Android* y la *Raspberry Pi* se comunican inalámbricamente a través de Internet WiFi.

8. Modelos estáticos y dinámicos de la solución de *software*

Una vez definido el hardware del sistema, se implementará una aplicación en cada uno.

El software de la *Raspberry* se va a dedicar a correr un sistema que detecte la acción de cierre y apertura de una caja fuerte (lo va a hacer mediante el sensor que tenga conectado a uno de sus puertos). A su vez, es capaz de enviar notificaciones a un celular por medio de la red y de desactivar localmente el sistema de seguridad presionando un botón.

A continuación se mostrará el modelo estático y dinámico que representa el software que correrá sobre la *Raspberry*:

[1]Para más información visitar `https://www.android.com/`

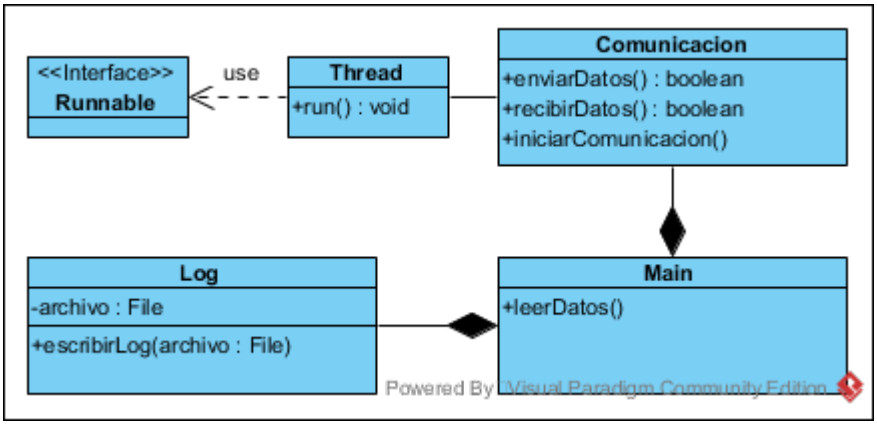

Figura 8: Diagrama de clases del sistema en la *Raspberry*.

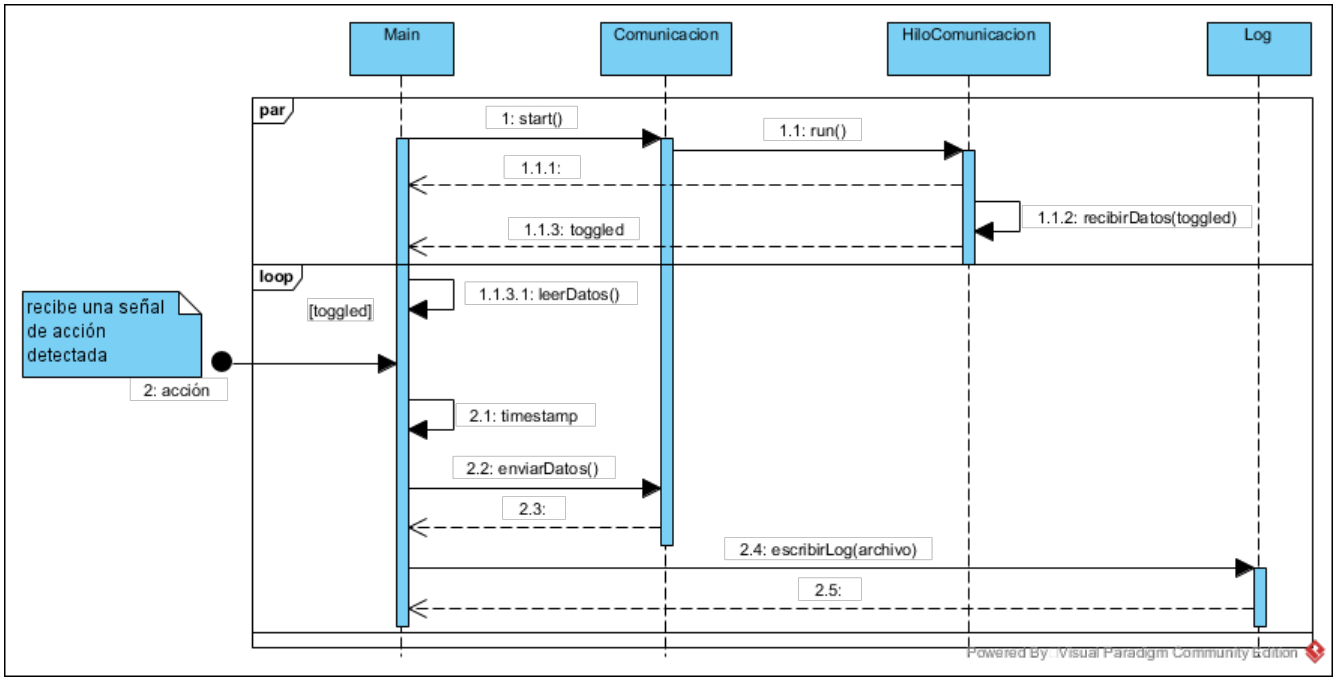

Figura 9: Diagrama de secuencia del sistema en la *Raspberry*.

El software del celular se va a dedicar a habilitar y deshabilitar en tiempo de ejecución de forma remota el sistema de seguridad de la caja fuerte. Para ello, por medio de la red debe mandar una señal que haga la orden.

A continuación se muestra el modelo estático y dinámico que representa el software que correrá sobre el celular *Android*:

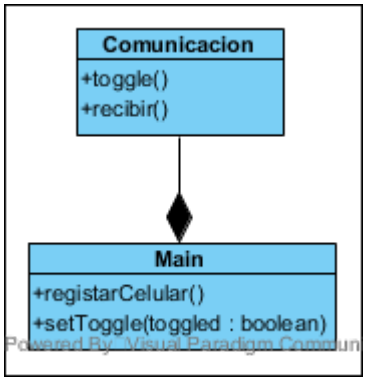

Figura 10: Diagrama de clases del sistema en el celular.

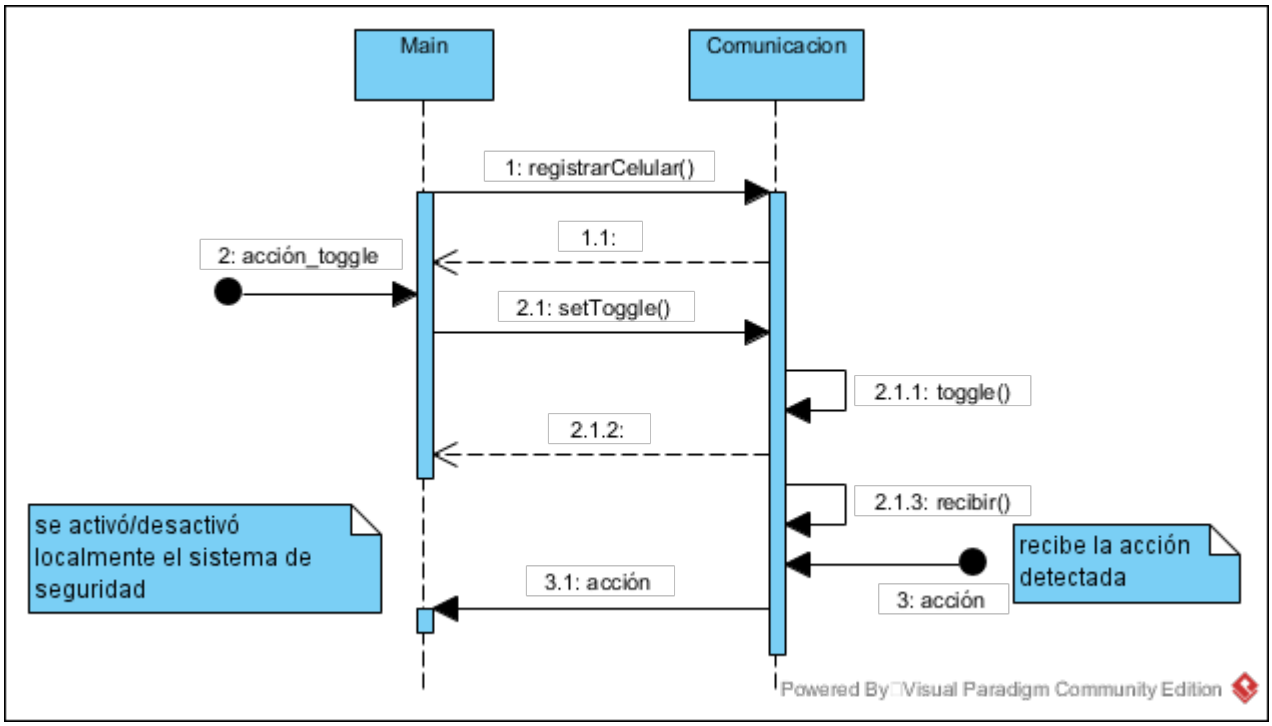

Figura 11: Diagrama de secuencia del sistema en el celular.

9. Descripción e implementación de simulación de entradas

Entrada	Salida
Cambio de estado de la caja fuerte	Agregar acción al *log* de eventos y notificarlo
Señal de activación del sistema	Sistema de seguridad activado
Señal de desactivación del sistema	Sistema de seguridad desactivado
Señal de registro	Se agrega un nuevo celular a la base de datos

Tabla 2: Casos de comportamiento del sistema.

10. Casos de prueba del *software/hardware*

En esta sección se muestran los casos de pruebas a realizar en el proyecto.

Código	**CP1**
Descripción	Registro de un nuevo celular
Pasos	▪ Abrir aplicación ▪ Presionar botón “Registrar celular"
Resultado esperado	El sistema debe haber agregado un nuevo celular a su base de datos y haber notificado el evento.

Tabla 3: Caso de prueba 1.

Código	**CP2**
Descripción	Desactivar sistema de seguridad remotamente
Pasos	▪ Abrir aplicación ▪ Presionar botón “Desactivar seguridad"
Resultado esperado	El sistema debe haberse desactivado y haber notificado el evento.

Tabla 4: Caso de prueba 2.

Código	**CP3**
Descripción	Activar el sistema de seguridad remotamente
Pasos	▪ Abrir aplicación ▪ Presionar botón “Activar seguridad"
Resultado esperado	El sistema debe haberse activado y haber notificado el evento.

Tabla 5: Caso de prueba 3.

Código	**CP4**
Descripción	Apertura y detección de la caja
Pasos	Abrir la caja fuerte
Resultado esperado	El sistema debe haber agregado este suceso al *log* y enviado la notificación correspondiente.

Tabla 6: Caso de prueba 4.

Código	**CP5**
Descripción	Cierre y detección de la caja
Pasos	Cerrar la caja fuerte
Resultado esperado	El sistema debe haber agregado este suceso al *log* y enviado la notificación correspondiente.

Tabla 7: Caso de prueba 5.

11. Conclusiones

El presente trabajo, nos propuso muchos y variados desafíos, desde cuestiones técnicas y de diseño a las que no estamos tan expuestos en la carrera, hasta el mero hecho de redactar un informe y todo lo que eso conlleva.

Respecto a esto último, el marco teórico fue lo primero que se llevó a cabo. Éste, enmarca al trabajo y al producto que pretendemos desarrollar dentro de un CPS, que al definirlo, nos brindó de un conjunto de herramientas que aplicamos en las fases siguientes de diseño.

Además, en el marco teórico se incurrió en el análisis de mercado del sistema de seguridad propuesto. Brevemente, éste nos permitió concluir que en nuestro país no existe producto alguno ni desarrollo del mismo.

Luego, se procedió a analizar cuáles eran los requerimientos del sistema que pretendemos desarrollar. Esto significó un claro entendimiento de que esperábamos que el prototipo haga, y a su vez, una minuciosa inspección del funcionamiento de este mismo, logrando extraer detalles que al ser tan comunes muchas veces se pasan por alto, por ejemplo obtener el *timestamp* de la acción realizada.

En la siguiente etapa, el labor de análisis permaneció vigente. A cada diagrama de bloques, de componente, o de contexto le precedió una detenida observación de qué bloques iban a intervenir en nuestro proyecto, cómo iban a interactuar y qué actores intervienen en el uso del mismo. Gracias a esto, se logró dar respuestas a estos interrogantes, y nuestro bloques resultaron ser el celular, el microcontrolador y la caja fuerte, mientras que los actores son únicamente dos: el controlador y el usuario.

Llegado este punto, se contó con la siguiente información para poder decidir en qué hardware se va poder concretar dicho diseño. Para el bloque del microcontrolador se eligió la *Raspberry Pi 3B+* dado que nos brinda grandes posibilidades de implementación y que ya se cuenta con dicha placa. Respecto al bloque del sensor, luego de pensar las alternativas posibles, se concluye que la más adecuada para la implementación de este prototipo es la unión de dos cables conductores. Por último, el celular del usuario es el mismo que actualmente utilizan los participantes del grupo.

Una vez determinado la elección del hardware a utilizar, se llevó a cabo la implementación de ciertos diagramas tanto estático como dinámicos para poder tener una visión más clara de cómo aplicar el uso de dichos componentes, es decir, con qué lógica van a interactuar entre ellos para poder cumplir con una mejor realización del proyecto. Esto se implementó tanto para la parte del microcontrolador como para el celular. Para poder determinar que esto se logró de forma exitosa se plantearon ciertos casos de pruebas.

Para concluir con la realización de este trabajo, se pudo dar una posible solución a la necesidad antes mencionada, de monitorear y conocer el estado de una caja fuerte sin tener que encontrarse físicamente en el mismo lugar. El diseño permite un agregado de funciones, como por ejemplo sensores de fuerza y presión, por lo que denota la escalabilidad de este proyecto.

Bibliografía

[1] Universidad de Oslo (2011), *An Introduction to Cyber-Physical Systems*, https://www.uio.no/studier/emner/matnat/ifi/nedlagte-emner/INF5910CPS/h11/undervisningsmateriale/20110830_CPS-WSN-Overview.pdf

[2] Infobae (08-08-2017), *El servicio privado de cajas de seguridad gana popularidad en argentina*, https://www.infobae.com/espacio-no-editorial/2017/08/08/el-servicio-privado-de-cajas-de-seguridad-gana-popularidad-en-la-argentina/

[3] Tecnosicurezza, https://www.tecnosicurezza.it/azienda.php

[4] Liz safe, http://www.lizsafe.com/

[5] AMSEC, https://www.amsecusa.com/about-cashwizard/product-information/

[6] SysML, https://omgsysml.org/

[7] Raspberrypi, *Raspberry Pi B+*, https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/