

Diciembre de
2017

Cerradura Inteligente

Proyecto PPS





Integrantes

- Quintana, Gabriel
- Di Maggio, Mariano



- Vivimos en un mundo híper conectado con Internet en constante evolución, con un crecimiento exponencial del uso de dispositivos para IOT.
- Se requieren soluciones ágiles para acompañar este crecimiento
- Hoy en día Internet ya no es un opcional, es un requerimiento obligatorio de muchísimas soluciones.
- Varias tecnologías acompañan este crecimiento y las mismas deben acompañar la demanda creciente





- El negocio detrás del producto tiene un crecimiento potencial.
- Infinidad de empresas han desarrollado sus sistemas de acceso mediante tarjetas RFID.
- Desde el punto de vista de la administración de usuarios se cuenta con una API rest centralizada en cloud.
- La API desarrollada se puede extender e implementar páginas web o aplicaciones de escritorio que la consuman.



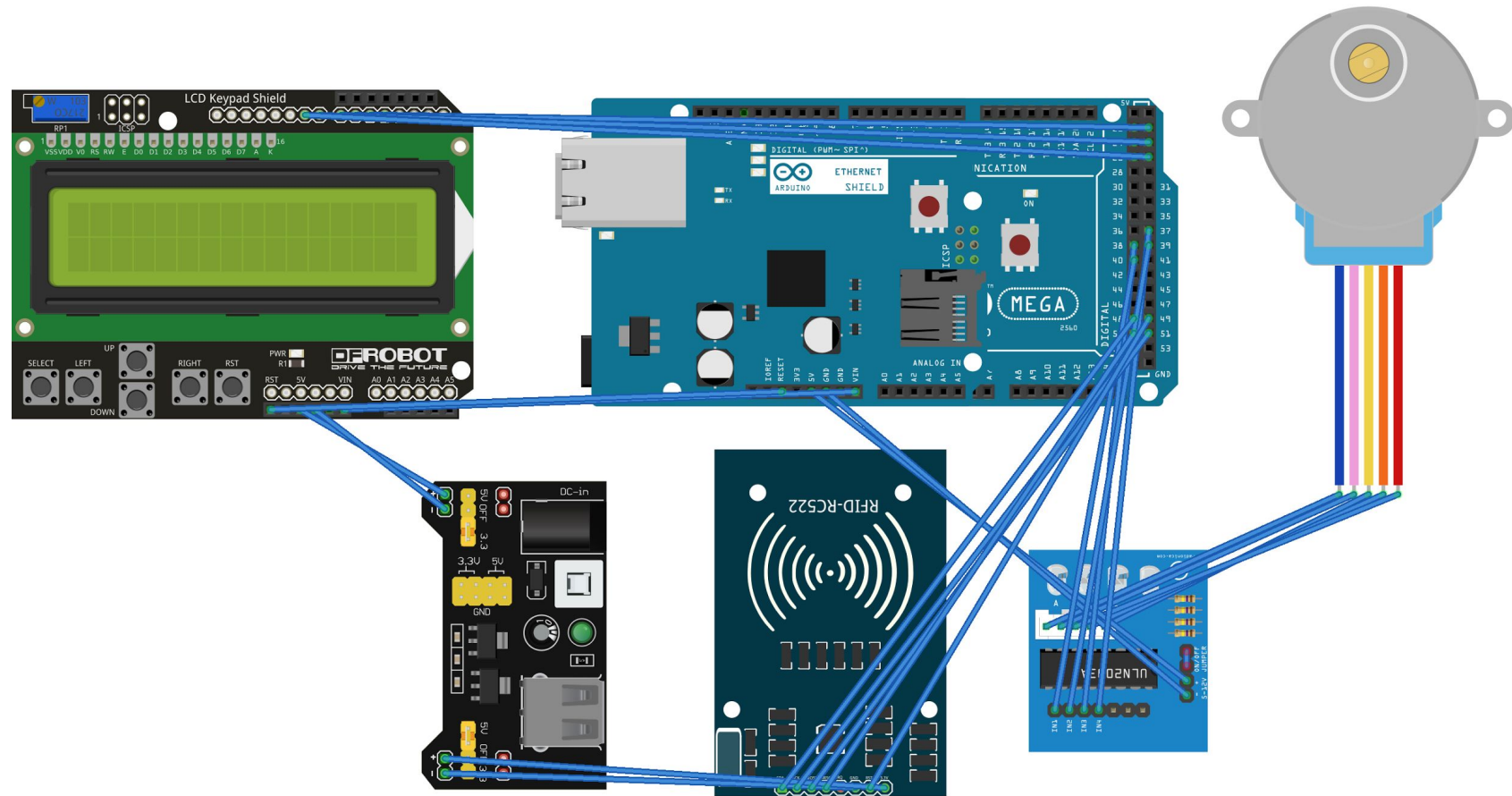


- Desde un punto de vista de costos del proyecto podemos hacer foco en dos elementos clave de la solución propuesta
1. Costos de desarrollo en Horas: El proyecto se desarrolló con tecnologías maduras y flexibles, que permiten una alta tasa de rendimiento y un bajo costo de mantenimiento.
 2. Las tecnologías que acompañan el desarrollo sobre dispositivos Arduino son económicas y abiertas.





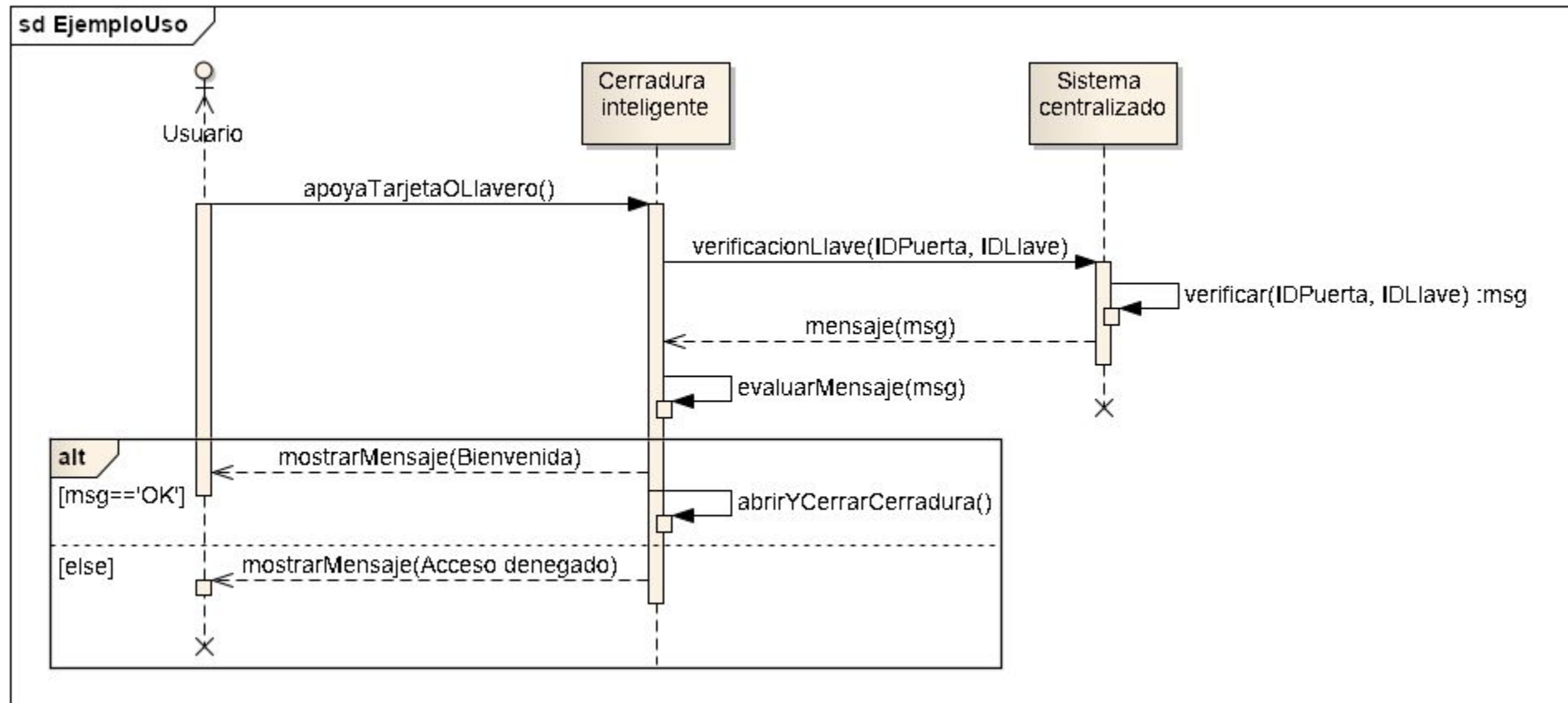
Componentes utilizados



fritzing



Funcionamiento básico:





```
void loop()
{
    if (mfrc522.PICC_IsNewCardPresent()) // Revisamos si hay nuevas tarjetas presentes
    {
        if (mfrc522.PICC_ReadCardSerial()) // Seleccionamos una tarjeta
        {
            client.print("GET /api/usuario/checkaccess/puertapublicidentification/");
            client.print(idPuerta);
            client.print("/llavepublicidentification/");
            for (byte i = 0; i < mfrc522.uid.size; i++)
            {
                if(mfrc522.uid.uidByte[i] < 0x10)
                {
                    Serial.print("0");
                    client.print("0");
                }
                Serial.print(mfrc522.uid.uidByte[i], HEX);
                client.print(mfrc522.uid.uidByte[i], HEX);
            }
            Serial.println();
            client.println();
            delay(1000);
            validarAcceso();
            Serial.println();
            client.stop();
            mfrc522.PICC_HaltA(); // Terminamos la lectura de la tarjeta actual
            verificarEstadoConexion();
        }
    }
}
```




Enumeramos algunas de las oportunidades de mejora del producto

1. Securitizar la conexión entre el Software Arduino y el sistema en Cloud.
2. Implementar un proceso alternativo en el caso que momentáneamente no se cuente con internet.
3. Crear un proceso de bloqueo que no implique gestiones administrativas complejas de parte del usuario (Administrador o usuario final).
4. Crear una UI, para facilitar la usabilidad del producto,
5. Posibilidad de crear grupos de puertas/llaves para facilitar la administración.
6. Enviar alertas por SMS e Emails .





Tecnologías utilizadas

- Arduino
- Lenguaje C
- Java EE SDK8 (JPA, Spring-data, Spring-mongo, Spring-kafka, etc)
- Spring Boot
- Swagger
- Kafka Broker/Kafka Tools/Zookeeper
- Docker/Docker-Compose
- MongoDB/Mongo-Express
- Github
- Maven





Metodología utilizada

Scrum

Componentes del equipo de proyecto

Jaguars

Product Owner

Profesores

Stakeholders

Equipo del Proyecto/ Profesores

Duración de los Sprint

Cada dos semanas con algunos sprint de 4 semanas

Daily

No teníamos daily, pero nos comunicamos de manera ágil mediante hangout





- Desarrollado para interactuar con elementos Cloud
- Componentes escalables (componentes stateless)
- Comunicación principal basada en Apis Rest
- Gestión de eventos mediante un motor de eventos asíncrono
- Parametrización flexible (Spring configuration)
- Facilidades para la gestión de acceso a datos (JPA)
- Compatibilidad con Docker u otros tipos de contenedores (Imagen oficial de Docker para MongoDB)
- Base de datos documental ideal para el manejo de datos de Redes Sociales y manejo de JSON , posibilidad de cambiar los datos y estructuras sin cambiar el código fuente.
- Facilidad para incorporar nuevas características





¿Qué?

Servicios que exponen lógica de Negocio, contienen la lógica, son el CORE de nuestra solución.

Interacción

- Interfaz de usuario
- Elementos de red

Lenguajes de programación

- Java

Frameworks

- Spring
- Spring boot
- Swagger

