ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

APLICACIÓN PARA CONTROL DE ACCESO MEDIANTE ARDUINO Y RFID

ACCESS CONTROL APLICATION USING ARDUINO AND RFID

Realizado por
Manuel Estepa Estepa
Tutorizado por
Daniel Garrido Márquez
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, junio 2016

Fecha defensa:
El Secretario del Tribunal

Resumen: En este Trabajo Fin de Grado se va a desarrollar una aplicación de ámbito empresarial encargada del control horario de los empleados de una empresa.

El sistema está formado por varios componentes. En primer lugar, de un subsistema hardware basado en microcontroladores Arduino que, haciendo uso de un lector de tarjetas RFID y una conexión inalámbrica a Internet, es capaz de enviar a un servidor web la identificación de quien ha pasado dicha tarjeta.

Este servidor web será el encargado de mostrar toda la información recogida por el subsistema hardware. Aquí, un empleado podrá ver las horas que lleva trabajadas en diferentes periodos de tiempo, así como el horario que tiene asignado. Finalmente, un administrador podrá comprobar las horas que todos los empleados de la empresa han trabajado, además de poder gestionar todo el sistema web (gestionar usuarios, instalaciones y horarios que más tarde se irán relacionando unos con otros) y gestionar notificaciones que los empleados hayan generado según sus necesidades.

Palabras clave: Arduino, Intel Edison, Java EE 7, JSF, EJB, JPA, HTML, XHTML, CSS RFID, PrimeFaces, BootsTrap, REST, Access Control.

Abstract: In this Final Degree Project will develop an aplication of Enterprise field. It will be responsible of the time control that an employee has been working in the company facilities.

The system consists of several components. First, a hardware subsystem based on Arduino microcontrollers, making use of a RFID card reader and a wireless Internet connection, it is able to send to a web server the identifying who has passed the card through the reader.

This web server will be responsible for displaying all information collected by the hardware system. Here, an employee will be able to see the hours he takes worked at different periods of time as well as the schedule is assigned for him. Finally, an administrator will be able to check the hours that all employees of the company have worked, and he will be able to manage all the web system (users, facilities and schedules management that later will relating to each other) and manage notifications employees have generated according to them needs.

Keywords: Arduino, Intel Edison, Java EE 7, JSF, EJB, JPA, HTML, XHTML, CSS RFID, PrimeFaces, BootsTrap, REST, Access Control.

ÍNDICE

1 INTRODUCCIÓN 9
2 PLANIFICACIÓN 13
3 ANÁLISIS DE REQUISITOS 15
3.1 Catálogo de usuarios 15
3.2 Requisitos funcionales 16
3.3 Requisitos no funcionales 17
3.4 Requisitos de información 17
4 ARQUITECTURA Y TECNOLOGÍAS 19
4.1 Hardware 19
4.2 Software 20
4.2.1 Nivel de cliente 22
4.2.2 Nivel de servidor Java EE 22
4.2.3 Nivel de base de datos 23
4.3 Comunicación 24
5 DISEÑO 25
5.1 Hardware 25
5.2 Software 29
5.2.1 Base de datos 29
5.2.2 Página web 31
5.3 Comunicación 38
6 DESARROLLO 39
6.1 Hardware 39
6.2 Software 40
6.3 Comunicación 47
7 PRUEBAS 49
8 LÍNEAS FUTURAS 51
9 CONCLUSIONES 53
10 REFERENCIAS BIBLIOGRÁFICAS 57

11 ANEXOS 59

11.1 Capturas de la página web 59

11.1.1 Páginas comunes 59

11.1.2 Rol de empleado 60

11.1.3 Rol de administrador 62

1 INTRODUCCIÓN

Tal y como se indica en el nombre de este proyecto (“APLICACIÓN PARA CONTROL DE ACCESO MEDIANTE ARDUINO Y RFID.”) la finalidad de éste es obtener como resultado una herramienta hardware y una aplicación web que cubran las necesidades que alguna empresa pueda tener para llevar a cabo el seguimiento de sus trabajadores, con el objetivo de saber cuánto tiempo pasan dentro de las instalaciones propiedad de la empresa.

Así, se ha desarrollado un sistema, con las especificaciones que se explicarán detalladamente más adelante, para cubrir este control de acceso con el que se han intentado cubrir los diferentes objetivos que se plantearon al principio del proyecto:

- Analizar, diseñar e implementar un sistema hardware para la lectura de tarjetas con tecnología RFID
- Analizar, diseñar e implementar una plataforma web donde se permita la visualización de la información recopilada por el sistema hardware. Este sistema a su vez planteaba diferentes subobjetivos:
    - Seguimiento de las horas trabajadas por los empleados.
    - Seguimiento de estancia actual en las instalaciones.
    - Posibilidad de generación de informes.
- Analizar, diseñar e implementar un sistema que se encargue de la correcta lectura de las tarjetas con tecnología RFID, así como la gestión de usuarios registrados en el sistema con una tarjeta.
- Analizar, diseñar e implementar un sistema de comunicación entre el sistema hardware y la plataforma web para poder transmitir los datos de un sistema a otro.

Como se explicará detalladamente en próximas secciones, algunos de estos objetivos se han alcanzado con combinaciones de soluciones aplicadas tanto en el sistema hardware como en el sistema software.

Tras exponer los objetivos del sistema, se va a nombrar las tecnologías más importantes que se han usado en este proyecto.

Para empezar, Arduino es una compañía de hardware libre, la cual desarrolla placas de desarrollo que integran un microcontrolador y un entorno de desarrollo (IDE), diseñado para facilitar el uso de la electrónica en proyectos multidisciplinarios.

En este proyecto se ha utilizado el microcontrolador Intel Edison con la placa de extensión correspondiente que aporta las mismas funcionalidades que los microcontroladores de Arduino, entre otras, pero que incorpora en su placa de extensión varios componentes electrónicos como son WiFi, Bluetooth o ranura para tarjetas SD. Estos componentes también pueden ser conectados individualmente a las placas proporcionadas por Arduino. Además, la placa Intel Edison puede ser también programada haciendo uso del entorno de desarrollo proporcionado por Arduino. Así, todo el desarrollo y especificaciones del hardware de este proyecto se hará basándose en el microcontrolador Intel Edison.

Otra de las tecnologías usadas en el proyecto es RFID (acrónimo de Radio Frequiency IDentification, en español identificación por radiofrecuencia). RFID es un sistema de identificación de objetos a distancia (puede ser variable) sin necesidad de contacto, ni siquiera visual, que usa dispositivos denominados etiquetas o tarjetas RFID cuyo objetivo es transmitir la identidad de un objeto mediante ondas de radio.

Por último, para el desarrollo de la plataforma web, que cubría algunos de los objetivos explicados más arriba, tras su análisis y diseño, se ha desarrollado haciendo uso de tecnologías actuales como son HTML5 y CSS3 para la creación de “front-end” de la página. Además, la interfaz se ha llevado a cabo mediante el uso del framework “Bootstrap”, lo que ha permitido que la página web sea “responsive”. Para la el “back-end” se ha hecho uso de la tecnología Java EE7, más exactamente haciendo uso de JSF (Java Server Faces) y un framework llamado “PrimeFaces”, framework que contiene una gran variedad de componentes muy potentes para el desarrollo de plataformas web.

Así, toda esta tecnología usada dará lugar a la aplicación general que se pretende conseguir como objetivo final.

Para la puesta a cabo del proyecto se han seguido diferentes etapas que estarán descritas detalladamente a continuación en el cuerpo de esta memoria. Las partes que se verán son:

- Planificación. Donde se comentarán todos y cada uno plazos que se enmarcan las diferentes fases del proyecto y que metodología se ha seguido.

- Análisis de Requisitos. Donde se estudiarán que requisitos comprenden el sistema completo, desde requisitos funcionales y no funcionales hasta requisitos de información. Además, se delimitará el alcance total del proyecto.

- Arquitectura y tecnologías. Donde se hablará de todas las tecnologías que se han usado para conseguir los objetivos y que arquitectura de desarrollo se ha seguido.

- Diseño. Donde se analizará el diseño que deba tener la aplicación según los requisitos estudiados anteriormente.

- Desarrollo. Donde se llevará a cabo la implementación de todo el sistema (hardware y software) según lo estudiado en los puntos anteriores.

- Pruebas. Donde hablará de como se ha comprobado el correcto funcionamiento de la implementación del sistema.

- Líneas futuras: Donde se hablará de posibles funciones que pudiesen ser añadidas al proyecto para mejorarlo.

- Conclusiones. Donde se comentará el resultado final del proyecto, los inconvenientes que se ha tenido durante todo el proceso, etc.

- Referencias bibliográficas.

- Anexos.

Así, basándonos en esta estructura, pasamos a conocer más a fondo cada una de las etapas del proyecto para conocer con exactitud cada uno de los elementos que lo integran.

2 PLANIFICACIÓN

En este primer apartado se pretende la planificación de todas las etapas a realizar para alcanzar el resultado del proyecto.

Así, este proyecto se ha realizado desde el mes de marzo de 2016 hasta junio del mismo año, periodo en el cual se plantearon diferentes objetivos a alcanzar. La planificación ha sido la siguiente:

- Especificación del sistema (análisis de requisitos): Antes del 10 de marzo.
- Diseño (donde se incluyó diseño de la aplicación y arquitectura a usar): Antes del 31 de marzo.
- Implementación (dividida en tres partes):
    - Desarrollo software (página web): Antes del 31 de mayo.
    - Montaje y desarrollo del hardware: Antes del 7 de junio
    - Implementación de la comunicación entre las partes: Antes del 7 de junio.
- Pruebas: Antes del 15 de junio.
- Documentación (esta misma memoria, la cual se ha ido haciendo poco a poco desde el inicio del proyecto hasta el final): Antes del 22 de junio.

Siguiendo esta planificación y teniendo en cuenta que la especificación se hará justo al inicio del proyecto, por lo cual se tendrán todos los requisitos que deberá cubrir el proyecto, se ha utilizado una metodología de desarrollo basada en el modelo en cascada, la cual es una de las metodologías que mejor se adaptan en los casos donde los requisitos son fijados desde un inicio del proyecto y donde no hay necesidad de cambios importantes.

3 ANÁLISIS DE REQUISITOS

En este apartado del proyecto, se analizará con rigurosidad los diferentes requisitos de diferentes tipos que cubrirán totalmente las necesidades del proyecto.

Por un lado, para conocer de mejor forma los posibles usos que se le darán al sistema, también se estudiarán los diferentes usuarios potenciales que podrían usarlo.

Así, se describirán tres tipos de requisitos:

- Requisitos funcionales.
- Requisitos no funcionales.
- Requisitos de información.

Con todo esto, quedará reflejado el alcance del sistema, el cual delimitará el proyecto.

3.1 CATÁLOGO DE USUARIOS O ACTORES:

En este apartado se analizarán cuáles serán los posibles tipos de usuarios que usarán el sistema, algo muy importante de cara a saber los posibles usos que un usuario desea del sistema.

Así, los diferentes posibles tipos de usuarios serían los siguientes:

- Empleado: Podrá observar su actividad y horario, además de generar notificaciones.
- Administrador: Podrá asumir también el rol de empleado, pero tendrá la posibilidad de gestionar los datos del sistema (gestionar empleados, instalaciones, horarios, etc.).

Pensando en los posibles casos de uso que tendrían estos usuarios, se deducen los requisitos que se muestran a continuación.

3.2 REQUISITOS FUNCIONALES:

En este punto se definirán las funciones que deberá cumplir el sistema de software y hardware o sus componentes. Para la identificación de cada uno de los requisitos se utilizará las siglas “RF” junto con un número identificativo para cada uno de estos.

Así, los requisitos funcionales son los siguientes:

- RF1: Gestión de lectura de tarjetas RFID.
    - RF1.1: Gestión de validación de tarjetas RFID.
    - RF1.2: Gestión de alta de tarjetas RFID en el sistema.
- RF3: Gestión de personal/usuarios (CRUD, Create, Read, Update, Delete).
- RF4: Control sobre la actividad en el sistema.
    - RF4.1: Control de acceso a la plataforma web.
    - RF4.2: Control de acceso a las instalaciones en tiempo real.
    - RF4.3: Control de acceso a las instalaciones de datos pasados.
- RF5: Gestión de informes sobre actividad. (CRU).
- RF6: Gestión de roles de usuarios.
- RF7: Control de comunicación entre componentes del sistema.
- RF8: Gestión de horarios.
    - RF8.1: CRU Horarios.
    - RF8.2: Asignación de horarios a empleados.
- RF9: Gestión de instalaciones (CRUD).
- RF10: Gestión de notificaciones (envío, recepción y resolución).

Así, con la cobertura de estos requisitos, los objetivos de este proyecto quedarían cubiertos.

3.3 REQUISITOS NO FUNCIONALES:

En este punto se definirán los requisitos que establecen límites al proyecto y que no dependen o establecen ninguna funcionalidad más al sistema. Así, al igual que anteriormente, se identificarán estos requisitos con las siglas RNF junto con un número identificativo para cada uno de estos.

Así, los requisitos no funcionales son los siguientes:

- RNF1: Sistema en tiempo real (hardware y software).
- RNF2: Facilidad de uso.
- RNF3: Almacenamiento de todos los datos necesarios.
- RNF4: Tiempo de respuesta del sistema reducido (hardware y software).
- RNF5: Soporte multiplataforma.

Así, todo el desarrollo deberá cumplir con estos requisitos para que el proyecto sea completo realmente.

3.4 REQUISITOS DE INFORMACIÓN

Por último, en el apartado de requisitos de información se describirán aquellas fuentes de información que deberán ser almacenadas para el correcto funcionamiento del sistema. Al igual que en apartados anteriores, quedarán identificados con las siglas RI junto con un número identificativo para cada uno de estos.

Así, los requisitos de información de este sistema son:

- RI1: Empleado. Se guardará la información personal del usuario, así como el rol que tendrá en la página web y su contraseña de acceso.
- RI2: Instalación: Guardará la información del tipo de instalación que se trata, ubicación, etc.
- RI3: Informe: Guardará los datos de quien ha generado el informe y una descripción de lo que contenía.

- RI4: Horario: Guardará los tramos horarios que podrán tener asignados los diferentes empleados que estén registrados en el sistema.

- RI5: Notificación: Se guardarán los datos de cuándo, quién y por qué se ha enviado dicha notificación para facilitar su resolución si es necesario.

- RI6: Accesos: Se guardarán los datos de quien ha accedido a alguna instalación y la fecha exacta de cuando ha ocurrido. Se registrarán tantos las entradas como las salidas a cualquiera de las instalaciones gestionadas por el sistema.

- RI7: Registro de acceso web (log): Se guardará la información sobre quien ha accedido a la página web, con qué rol y en qué momento para su supervisión.

Por lo tanto, con estos requisitos de información, el sistema podrá funcionar haciendo uso de todos los datos anteriormente descritos.

Así, con todo este catálogo de requisitos se pretende cubrir los objetivos planteados en la introducción de este proyecto. Sobre todos los requisitos se basará el diseño y arquitectura del sistema que se llevará a cabo en los próximos apartados, en los cuales, haciendo uso de diferentes casos de uso, máquinas de estado, y otros recursos, se pretenderá plantear el sistema de manera inequívoca y exacta para su posterior implementación en la fase de desarrollo.

4 ARQUITECTURA Y TECNOLOGÍA

En este apartado se describirá que arquitectura se ha seguido para el desarrollo de todo el sistema según las especificaciones hechas en los requisitos y de forma paralela al diseño del sistema. Además, se explicarán los lenguajes con los que se ha hecho el desarrollo junto todas las tecnologías y frameworks usados para una mejor solución.

Teniendo esto en cuenta se hablará de las tres partes que diferencian el sistema: hardware, software y comunicación entre ambos sistemas.

4.1 HARDWARE:

Por su parte, el hardware se ha basado en dos tecnologías diferentes: Arduino (Intel Edison) y RFID.

Arduino es una empresa que se encarga del desarrollo de microcontroladores con diferentes placas de expansión para ofrecer diferentes tipos de prestaciones. Arduino en su versión UNO permite la comunicación serie además de contener 16 pines para conectar componentes digitales y 6 pines para la conexión de componentes analógicos.

En este proyecto se van a usar algunos leds, algunos botones y un buzzer, todos necesitan conexiones con pines digitales. Esto, unido a los pines necesarios para el módulo RFID, deja pocos pines libres para su uso.

Por esto, debido a que todavía habría que conectar el módulo de conexión inálambrica (WiFi) se ha hecho uso de Intel Edison que, como ya se adelantó algo en la introducción, puede utilizarse de forma análoga a Arduino.

Intel Edison nace como proyecto para dar un impulso al IoT (Internet of Things, es español Internet de las cosas). Así su placa de extensión contiene muchos de los componentes que podrían usarse en Arduino de manera individual como pueden ser el módulo WiFi o el módulo Bluetooth. Además, uno de los objetivos de Intel Edison es que el desarrollo de sus programas pudiese adaptarse fácilmente a Arduino, por lo que permite ser programada haciendo uso del entorno de desarrollo (IDE) de Arduino.

La otra tecnología importante en lo que se refiere al hardware será RFID (Radio Frequency IDentification). Ésta es una tecnología que se basa en la comunicación a corto alcance sin necesidad de contacto haciendo uso de etiquetas RFID.

Las etiquetas RFID son unos dispositivos pequeños, similares a una pegatina, que puede ser adheridos a diferentes objetos con el fin de poder darles una identificación. Estas etiquetas pueden ser pasivas, activas o semiactivas dependiendo de cuándo y cómo funcionan. En este proyecto se utilizarán etiquetas pasivas, las cuales necesitan de un lector RFID cercano para se activen y manden la información de identificación del objeto.

El montaje se realizará siguiendo los esquemas que se especificarán en el diseño hardware, usando y conectando con exactitud los componentes descritos en dicho apartado que se encontrará a continuación.

La arquitectura seguida para el desarrollo de una aplicación basada en Arduino es la que se comenta a continuación.

La programación del sistema hardware se basará en la arquitectura que tiene toda aplicación desarrollada para Arduino, o derivados como es este caso con Intel Edison. Así, como se plantea en la base de todo proyecto realizado con el entorno de desarrollo de Arduino, todo desarrollo se basa en dos componentes: “setup” y “loop”.

En “setup” se inicializan todos los pines y componentes que se van a usar en el sistema (leds, botones, lector RFID, conexión WIFI, etc) y en “loop”, como su propio nombre indica, se ha implementará lógica que se especificará en el diseño, algo que se irá iniciando una vez tras otra mientras que el sistema esté conectado.

4.2 SOFTWARE:

En lo que se refiere a software, el uso del lenguaje Java EE 7 (Enterprise Edition) nos proporciona una serie de herramientas para el desarrollo de aplicaciones “empresariales”. Estas herramientas hacen que la arquitectura del software se divida en varias capas muy diferenciadas. Estas capas deben ser implementadas de forma bastante diferente teniendo en cuenta el rol que juegan cada una de ellas.

Las capas en las que se divide la una aplicación según Java EE 7 son:

- Cliente (navegador web).
- Web (página desarrollada).
- Servidor Java EE.

- Sistema de información.

Sin embargo, de cara al desarrollo de aplicaciones, suelen tenerse en cuenta las siguientes capas, aunque como es obvio tienen una gran relación las unas con las otras (arquitectura que se ha seguido en este proyecto):

- Cliente (página web).
- Servidor Java EE.
- Base de datos.

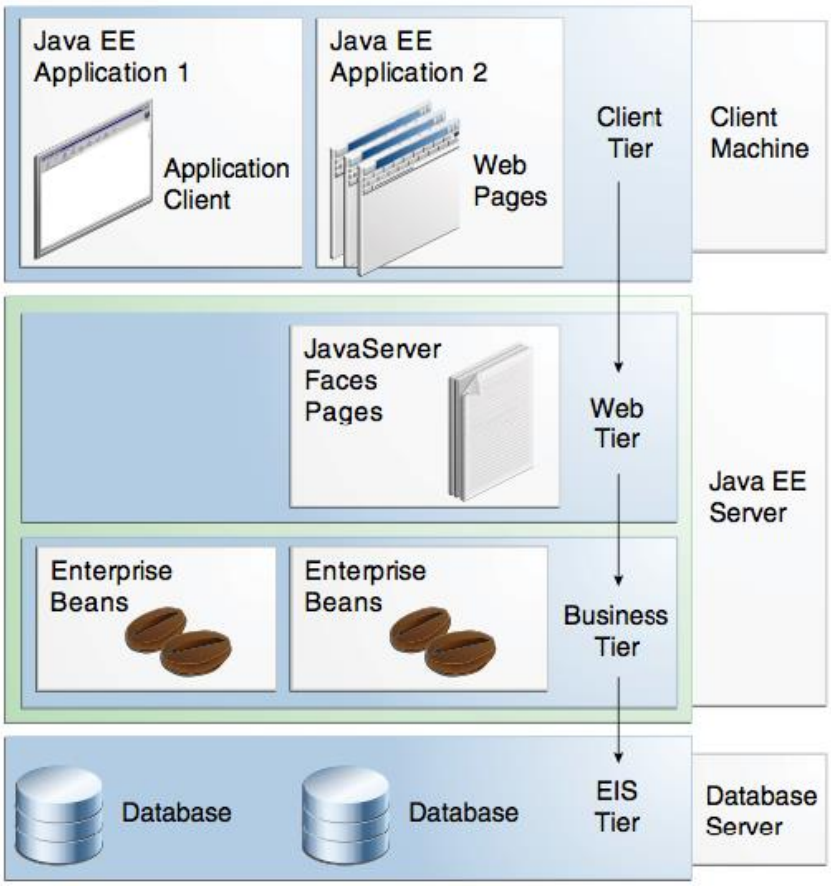


Figura 1: Imagen arquitectura Java EE.

Así, en cada uno de los niveles o capas se han usado tecnologías diferentes, unas directamente relacionadas con programación Java y otras complementarias para este proyecto. A continuación, se explicarán en cada nivel que lenguajes o tecnologías se han usado para su implementación, así como una pequeña descripción de cada uno de los niveles.

- 4.2.1 Nivel de cliente:

  En este nivel se desarrollan todas y cada una de las páginas web a la que un usuario podrá acceder, por esto es conocida como capa de presentación. Por lo tanto, para esto, el usuario deberá poseer en su equipo un navegador web para acceder a dichas páginas.

  Así, para su desarrollo se han hecho uso de los lenguajes HTML5 (HyperText Markup Language en su versión 5) y CSS3 (Cascading Style Sheets en su versión 3). Estos dos lenguajes se han usado para el desarrollo de la interfaz de usuario. Además, se ha hecho uso del framework Bootstrap para hacer que la página web sea "responsive", es decir, que la interfaz de usuario se adapte a todas las resoluciones que pueda tener la pantalla, permitiendo que, al acceder desde un dispositivo móvil, por ejemplo, la interfaz siga teniendo un diseño agradable.

  Por último, comentar el uso de AJAX (Asynchronous JavaScript And XML) para facilitar la forma de actualizar componentes de una página web sin necesidad de actualizar la página al completo.

- 4.2.2 Nivel de servidor Java EE:

  Por un lado, en este nivel se incluye la tecnología Java denominada JSF (Java Server Faces), la cual permite dar funcionalidad a las páginas web creadas con las tecnologías anteriores. Para el uso de JSF ha sido necesario la modificación de todas las páginas hechas en HTML a XHTML (eXtensible HTML), el cual es necesario para la comunicación con las clases Java creadas usando JSF. Añadir, que para mejorar la potencia que ofrece JSF, se ha usado el framework PrimeFaces, el cual añade y mejora los componentes que nos ofrece JSF por defecto.

  Por último, decir que JSF ofrece varios contextos de funcionamiento conocidos como "backing beans", que son la interfaz con la capa de negocio, que se explicará a continuación y que determinan cómo va a ser la navegación, además de contener los datos a mostrar en la vista, así, estos deben ser clases POJOs. Estos "backing beans" pueden tener varios ámbitos, o más conocidos por la terminología inglesa "scoped". Los "scoped" usados en este proyecto han sido de dos tipos principalmente, "SessionScoped" y "ViewScoped".

  - SessionScoped: El "bean" existirá durante una sesión, por lo tanto, se ha usado para guardar los datos de una sesión de usuario como puede ser el caso de los datos del usuario que ha iniciado sesión.

- ViewScoped: El “bean” existirá mientras el usuario se encuentre en la vista relacionada, por lo que se han usado para guardar los datos necesarios para el correcto funcionamiento de cada una de las vistas de la página web.

Por otro lado, en la subcapa conocida como capa de negocio, se usará otra tecnología Java conocida como EJB (Enterprise Java Beans).

Esta capa será la encargada de conectar la base de datos con el resto de la arquitectura anteriormente comentada. Estos “beans” pueden ser de tres tipos: sin estado (stateless), con estado (stateful) y singular (singleton). Así, en este proyecto sólo se han usado beans del tipo “staleless”.

- 4.3.3 Nivel de base de datos:

En este nivel, como su propio nombre indica, es donde se realiza la declaración de la base de datos para poder accedida por el resto de la aplicación.

Para su desarrollo se ha utilizado la tecnología java JPA (Java Persistence API). Esta tecnología permite al desarrollador ver la estructura de una base datos basada en objetos, algo conocido como ORM (Object-Relational Mapping).

Así, cada entidad de la base de datos se ha desarrollado como un objeto, implementada como clase de tipo POJO, en la cual cada atributo de la entidad se declara como una variable java con el tipo que se crea conveniente.

Por otro lado, las relaciones entre cada una de las entidades se han realizado haciendo uso de algunas anotaciones proporcionadas por JPA, lo cual hace que una relación puede declararse con el mero hecho de definir en ambas entidades relacionadas un objeto o una lista de objetos de estas clases dependiendo de la cardinalidad de ambas partes de la relación.

Por lo tanto, JPA permite crear una base de datos haciendo uso de clases en Java que luego serán mapeadas como una base de datos relacional en el servidor. Esta base de datos es accedida desde la capa de negocio haciendo uso de sentencias basadas en el lenguaje JPQL (Java Persistence Query Language).

Toda esta arquitectura basada en Java EE 7 ha sido desplegada sobre un servidor GlassFish en su versión 4.1. GlassFish es un servidor de aplicaciones de software libre desarrollado por “Sun Microsystenms”, una compañía de “Oracle”. Implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

4.3 COMUNICACIÓN:

Para la comunicación entre los sistemas hardware y software se usará un servicio web basado en la tecnología REST (Representational State Transfer, o en español Transferencia de Estado Representacional), también conocido como RESTful.

REST se basa en un conjunto de operaciones básicas bien definidas. Las operaciones principales son POST, GET, PUT y DELETE. Estas cuatro operaciones representan lo que en base de datos es conocido como CRUD (Create, Read, Update y Delete, o en español Crear, Leer, Actualizar y Borrar).

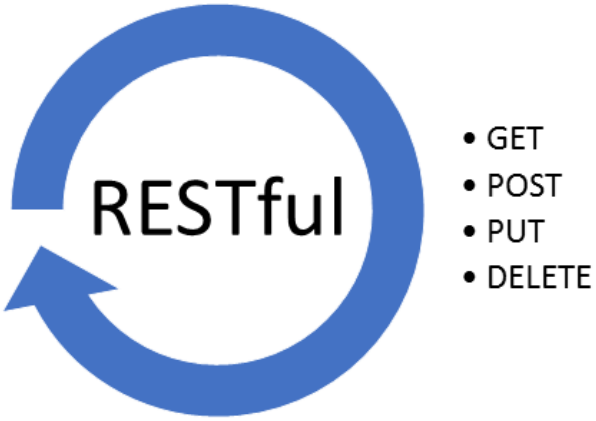


Figura 2: Imagen representativa servicios web REST.

Así, desde programación Arduino se deberá usar este tipo de comunicación con el servidor, de forma que se haga una petición POST con la información necesaria para ser incluida en la base de datos (en este caso la tarjeta RFID). Tras esto, en el servidor se tratará la información para ser almacenada en la base de datos en la correspondiente tabla obteniendo quien es el propietario de dicha tarjeta y cuándo se ha pasado la tarjeta por el lector.

5 DISEÑO

En este apartado se estudiará el diseño que deberá cumplir el sistema para cubrir con todos los requisitos descritos en el apartado anterior. Así, habrá que diferenciar las tres partes que componen el sistema: hardware, software y comunicación entres amos sistemas.

5.1 HARDWARE:

Como ya se adelantó, el sistema hardware será el encargado de la lectura de las tarjetas RFID que deberán llevar cada uno de los empleados para registrar en el sistema cuando han entrado o han salido de las instalaciones.

Para esto, se ha creado una circuitería electrónica basado en microcontroladores Arduino. Sin embargo, para la reducción de componentes electrónicos externos, como puede ser un módulo de conexión inalámbrica (WiFi), se ha preferido usar un microcontrolador denominado Intel Edison como ya se adelantó anteriormente. Como se puede observar en la próxima imagen, ambas placas de extensión contienen los mismos pines análogicos y digitales para conectar diferentes componentes.

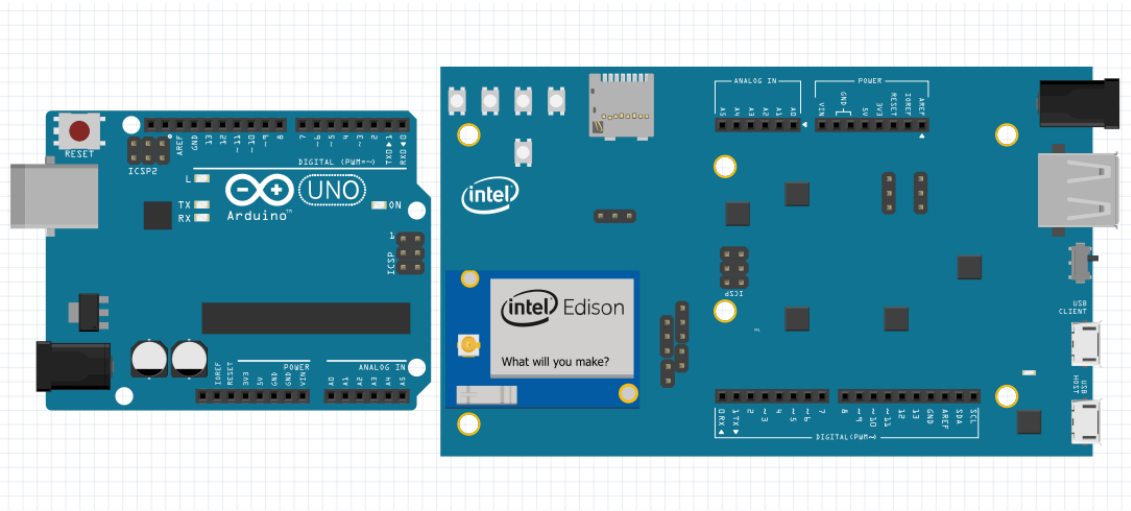


Figura 3: Comparativa Arduino Uno – Intel Edison.

Por lo tanto, este desarrollo hardware pudiera haberse llevado a cabo haciendo uso de Arduino en su versión UNO, pero para reducir la circuitería externa, se tomó la decisión de usar Intel Edison.

Tras la decisión de que controlador usar, el siguiente paso será ver que componente permite las lecturas de tarjetas RFID. El componente electrónico que permite estas acciones es conocido como RFID-RC522 el cual, haciendo

uso de las comunicaciones SPI proporcionadas por las librerías de Arduino, permite la lectura de tarjetas RFID.

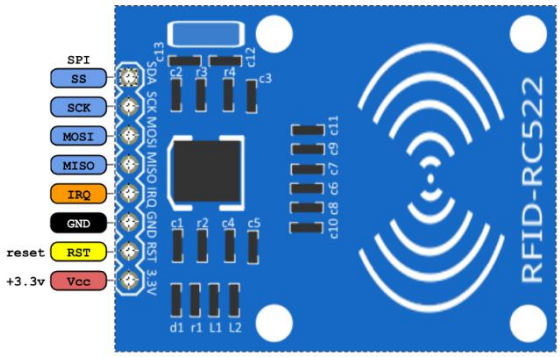


Figura 4: Módulo lector RFID.

Por lo tanto, conectando cada uno de los pines del módulo RFID a su correspondiente pin en la placa Intel Edison y su correspondiente programación permitirá la lectura de tarjetas. Las conexiones deben hacerse siguiendo la siguiente tabla:

Pines Intel Edison	Pines Módulo RFID-RC522
10	SDA(SS)
13	SCK
11	MOSI
12	MISO
No conectado	IRQ
GND	GND
9	RST
3.3V	3.3V

Después de conectar el módulo RFID a Intel Edison, había que añadir al sistema la forma de diferenciar cuando se estaba entrando o saliendo de una instalación por lo que se decidió añadir dos botones al sistema, uno para entrada y otro para la salida. Así, el botón de entrada (necesitando una resistencia de 10 KΩ cada botón) fue conectado al pin 6 y el de salida al 7.

Junto a esto era necesario la notificación de diferentes avisos por medio de componentes electrónicos, por lo que se ha hecho uso de tres leds y un piezo buzzer para emitir diferentes sonidos. Estos cuatro componentes (necesitando resistencias de 220 Ω cada uno) fueron conectados de la siguiente forma: led azul al pin 2, led rojo al pin 3, led verde al pin 4 y buzzer al pin 5.

Todo este circuito queda reflejado en la siguiente imagen.

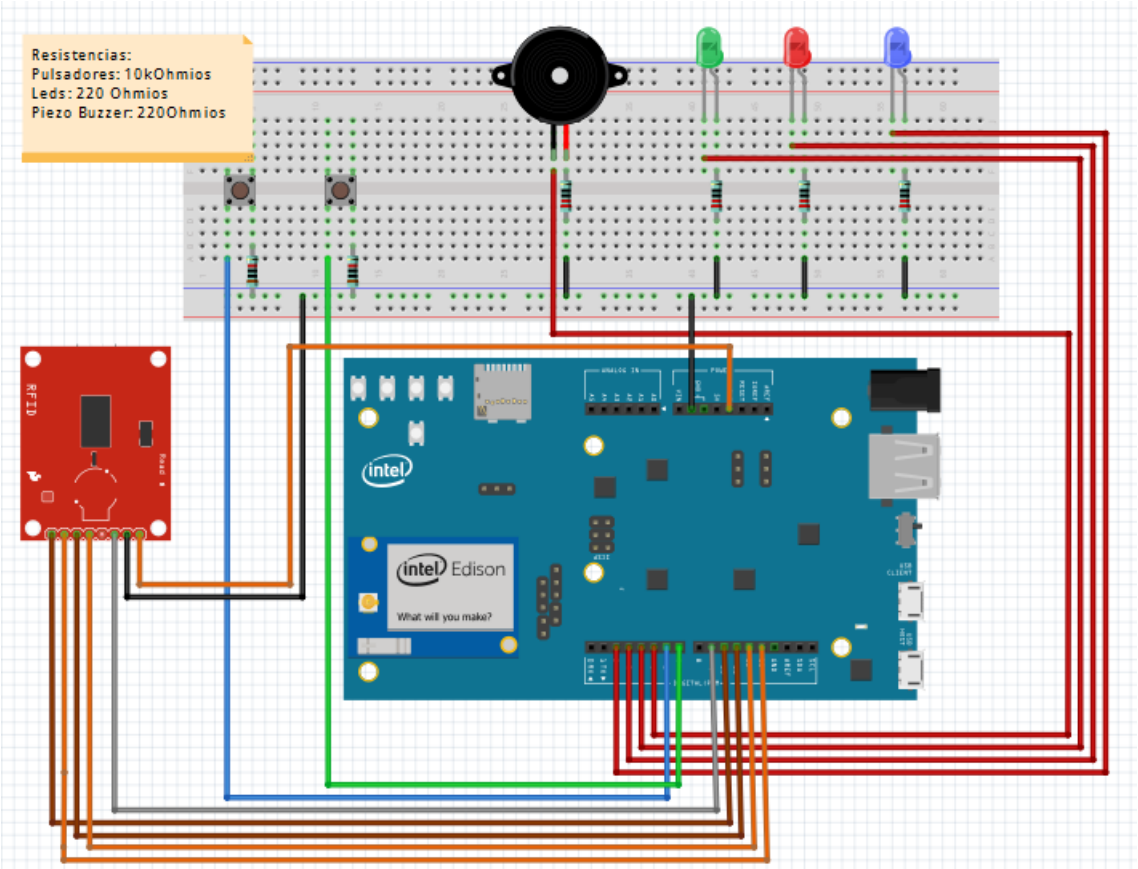


Figura 5: Esquema circuito sistema hardware.

Teniendo en cuenta que el módulo WiFi necesario está incluido en la placa de Intel Edison, sólo quedaría saber cómo debería funcionar todo el sistema hardware para su posterior correcta implementación. Esto queda completamente reflejado en el siguiente diagrama de estados:

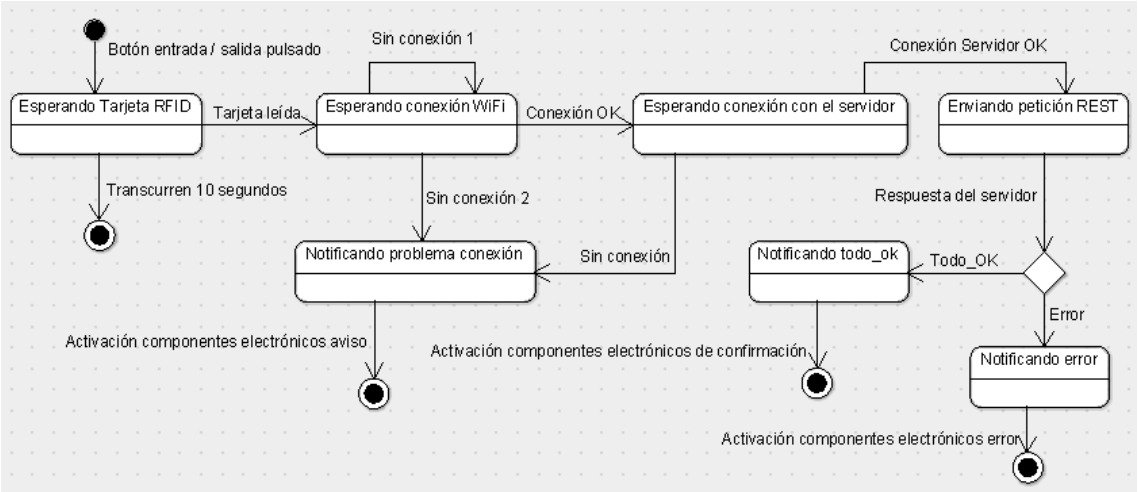


Figura 6: Máquina de estados. Sistema Hardware.

Algunas consideraciones a tener en cuenta en el anterior diagrama de estados son:

- En el punto inicial el led azul debe estar activado para indicar que el sistema está en su situación inicial por defecto.

- Tras salir del estado inicial el led azul debe estar desactivado lo que indicará que se está realizando alguna acción.

- Cada uno de los puntos finales se corresponderían con la vuelta al estado inicial.

- La activación de componentes electrónicos de confirmación se corresponde con la activación del led verde durante un segundo y del buzzer durante medio segundo concurrentemente.

- La activación de componentes electrónicos de error se corresponde con la activación del led rojo durante dos segundos y del buzzer durante un segundo y medio concurrentemente.

- La activación de componentes electrónicos de aviso se corresponde con la activación y desactivación tres veces cada medio segundo del led rojo y del piezo.

- Por último, señalar uno de los estados es “Esperando conexión con el servidor”, refiriéndose al servidor donde estaría alojada la base datos y la plataforma web.

Con este diagrama de estados queda totalmente explicada la funcionalidad que debería tener el sistema y junto con el diseño de la circuitería, también expuesto más arriba, quedaría el sistema hardware totalmente especificado.

Tras esto, el único uso que tendría este sistema sería el de registrar una entrada o una salida en la base de datos. Esto queda reflejado en el siguiente caso de uso.

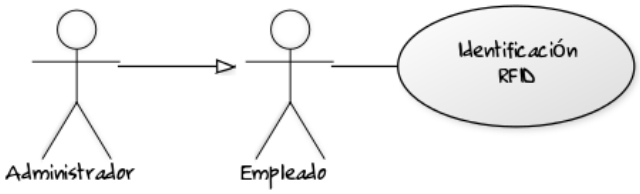


Figura 7: Diagrama de casos de usos. Sistema Hardware.

Esta identificación RFID sólo constaría de la necesidad de pulsar el botón de entrada o salida y de pasar la tarjeta RFID por el lector. Existe el posible problema de que el sistema falle y se deba avisar de esto a través de la página web, pero es algo que queda fuera del sistema hardware y que por tanto se verá posteriormente en su correspondiente apartado.

5.2 SOFTWARE:

Tras haber completado el diseño del sistema hardware, el sistema software que lo complementará y completará el sistema final será una página web con diferentes funcionalidades que cubrirán la totalidad de requisitos descritos en apartados anteriores.

En lo que se refiere a software, el sistema puede ser divido en dos partes más, debido a la diferencia sustancial que hay a la hora de especificar su diseño. Estas partes serán el diseño de la base de datos y el diseño de la página web.

5.2.1 Base de Datos:

La base de datos de este sistema parte de la necesidad de guardar la información de, al menos, todos aquellos requisitos de información descritos en su correspondiente sección.

Para el diseño de dicha base de datos se utilizará un modelo Entidad/Relación que permitirá conocer como deberá implementarse dicha base de datos. Este modelo queda reflejado en la siguiente imagen.

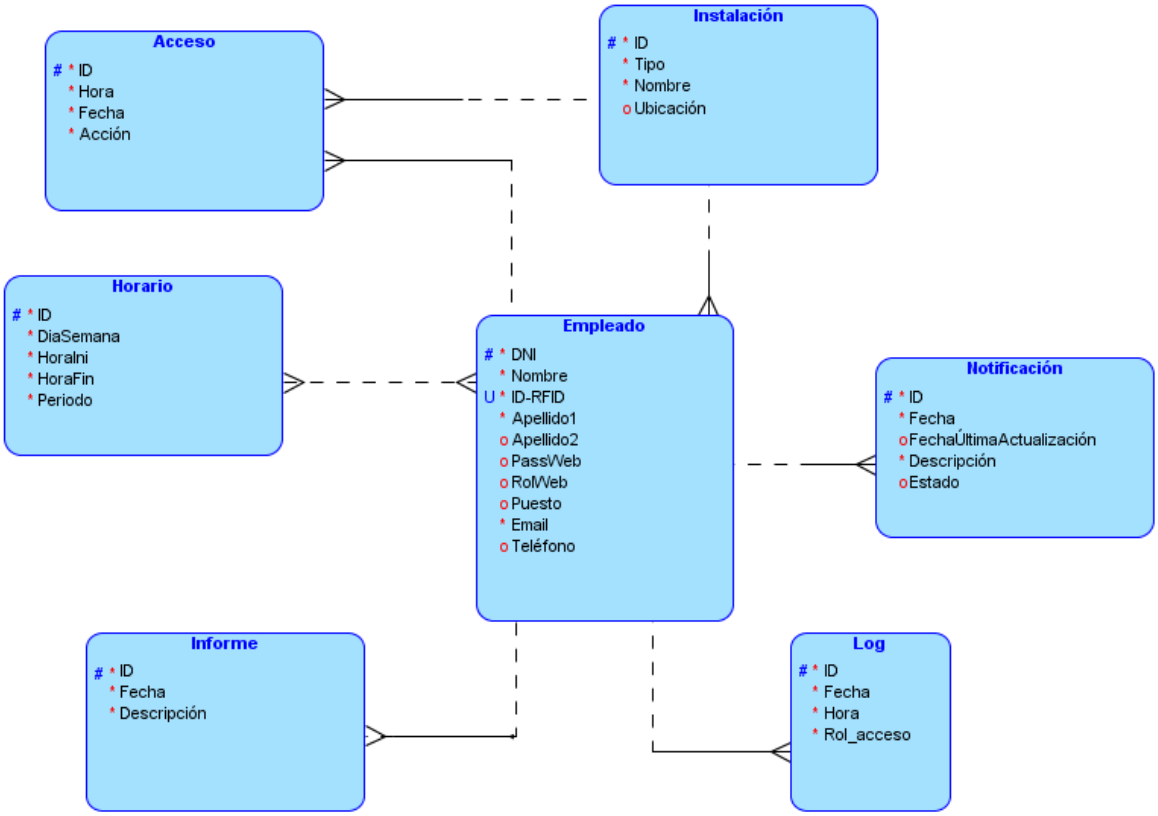


Figura 8: Diagrama entidad/relación. Base de datos.

Observando la imagen anterior, se podrán observar las diferentes relaciones que se han establecido y se puede ver como el empleado es el centro de toda la información que se guardará en la base de datos. Algunas consideraciones a tener en cuenta en el anterior modelo entidad/relación son:

- Líneas discontinuas implican que la relación no es obligatoria en esa dirección y líneas continuas implican que si lo es.

- El símbolo “#” indica que ese atributo será clave única e identificador de la entidad a la que pertenece. El símbolo “U” indica que ese atributo debe ser único en cada registro y no debe repetirse. El símbolo “*” indica que el atributo es obligatorio. El símbolo “o” indica que el atributo no es obligatorio.

- Las relaciones que contienen en alguno de sus dos lados un “muchos” quedan reflejados con el símbolo de las tres rallas

Con esta especificación de la base de datos, todos los requisitos de información quedan cubiertos y, por lo tanto, toda la información que se manejará en el sistema podrá ser almacenada y podrá ser accedida a través de la página web, cuyo diseño se describirá a continuación.

5.2.2 Página Web:

Tras la descripción de la página web que hará uso de la base de datos anteriormente descrita, ésta estará determinada por todas aquellas funcionalidades que un usuario, ya sea empleado o administrador, podrá usar para conocer la actividad que se está registrando.

Para este caso, la descripción de la funcionalidad estará descrita por diferentes casos de uso que los usuarios puedan dar al sistema.

El primero de todos, será el simple hecho de iniciar sesión en la página web, para esto se deberá introducir DNI y contraseña, además de seleccionar el rol con el que el usuario desea acceder. Habrá que tener en cuenta que el usuario se equivoque al introducir alguno de los datos, por lo que se mostrarán diferentes mensajes de aviso sobre cualquier ocurrencia.

Por otro lado, todo usuario podrá recuperar su contraseña, o más bien generar una nueva automáticamente. Para esto deberá introducir su DNI y correo electrónico asociado al sistema para enviar los datos necesarios a su email.

A continuación, se muestra el caso de uso que describe estos funcionamientos:

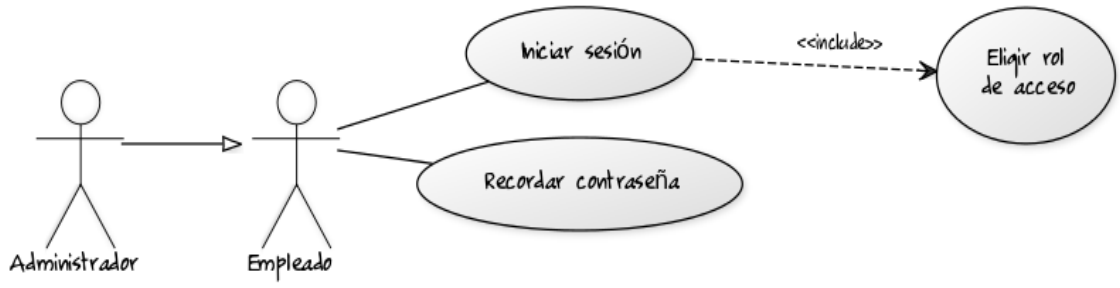


Figura 9: Diagrama de caso de uso. Inicio de sesión.

Tal y como se puede observar, los diferentes actores que podrán hacer uso de la página web serán o empleados o administradores. Decir que un administrador puede jugar también el papel de un empleado para ver su información personal solamente. Por esto, en todos los casos de uso que un empleado pueda realizar alguna acción, el administrador podrá jugar su papel.

Tras esta anotación, una vez que un usuario inicia sesión con el rol de empleado, éste podrá ver al inicio los datos personales que tenga asociados a la empresa. Estos datos pueden ser el DNI, nombre y apellidos, teléfono, email, rol web, puesto, etc. Algunos de estos datos, como son el email y teléfono podrán ser modificados directamente por el empleado sin necesidad de consultar con un administrador.

Otro uso que un empleado podrá hacer será ver el horario que tiene asignado. Este horario se separará en dos tablas para diferenciar el horario que tendrá en el periodo de mañana y en el periodo de tarde.

También el usuario podrá ver las horas que lleva trabajadas, lo cual necesita de la más fiel diligencia de los empleados a la hora de hacer pasar sus tarjetas RFID por el lector del sistema hardware, ya que no de ser así puede que las horas que se muestren en el sistema no sean las que realmente el empelado ha trabajado. Por esto se ha pensado en usar diferentes mensajes en esta página para que el empleado conozca cuales son los tramos horarios donde, ya sea a la entrada o a la salida de alguna instalación, no haya pasado su tarjeta por el lector.

Así, para poder conocer las horas que ha trabajado, el empleado deberá filtrar las horas trabadas por diferentes periodos preestablecidos: Hoy, Ayer, Esta semana, Este mes y El mes pasado. Todos los datos deberán aparecer en una tabla con la fecha, hora inicio, hora fin y suma de horas en caso de ser alguno de los dos primeros periodos y sólo fecha y total de horas en caso de ser alguno de los tres periodos restantes. Además, podrá observar la cuenta del total de horas del periodo filtrado.

Se dará la opción de que el empleado pueda obtener en formato PDF los datos que se hayan generado con esta búsqueda. Esta acción deberá quedar reflejada en la base de datos sobre quién, cuándo y sobre qué se ha generado dicho informe.

Otra de las opciones que deberá tener el empleado es la de poder gestionar sus notificaciones. Esta funcionalidad se encuentra de cara a que un administrador pueda ser notificado de cualquier incidencia, como puede ser la de cambiar algún dato personal que el empleado no pueda modificar sin permiso o el simple hecho de notificar a un administrador que ha olvidado o ha tenido algún problema a la hora de pasar su tarjeta por el lector.

Así, para la creación de una notificación el empleado deberá simplemente introducir una descripción de lo que pretende con esta notificación para que más tarde algún administrador pueda resolver el problema. A la hora de ser enviada la notificación al sistema, se deberá notificar por email al menos a un

administrador para que este cuanto antes pueda resolver cualquier problema que se haya podido ocasionar.

Por otro lado, el empleado podrá observar todas las notificaciones que ha generado para conocer su estado y así poder estar al día de la resolución de éstas. Esto lo podrá hacer con una tabla donde aparecerá en qué fecha se envió, que descripción incluyó y el estado exacto. Se dividirá la vista para ver las notificaciones que estén en estado pendiente o resueltas.

Por último, un empleado podrá cambiar de rol siempre y cuando éste tenga el rol de administrador, por lo cual se deberá mostrar un mensaje informando cuando el empleado no pueda realizar este cambio de rol.

Y, para terminar, el usuario podrá en todo momento cerrar sesión, lo que hará que el usuario vuelva a la primera vista de Inicio de sesión.

Todos estos posibles usos que un empleado podrá hacer en la aplicación. Esto queda reflejado en el próximo diagrama de casos de uso en el cual, tal y como se adelantó, un administrador podría jugar el papel de empleado para ver su información personal.

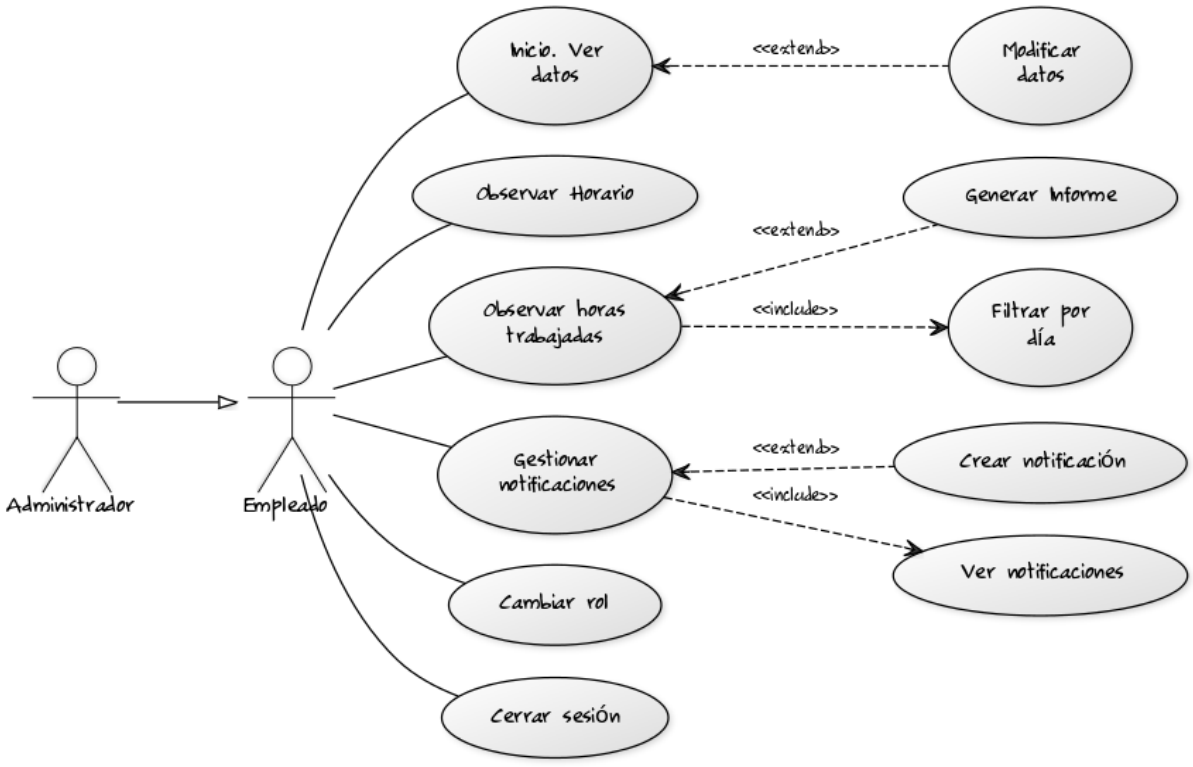


Figura 10: Diagrama de caso de uso. Funcionalidades del empleado.

Si el usuario registrado tiene permisos para entrar con el rol de administrador y elige esta opción a la hora de iniciar sesión tendrá otras funcionalidades

diferentes a su disposición, las cuales están totalmente relacionadas con las que puede desempeñar un usuario con el rol de empleado.

Así, el administrador lo primero que podrá observar serán sus datos personales al igual que el empleado, exactamente con los mismos datos.

La siguiente funcionalidad a la que debe poder acceder el administrador deberá ser una vista en la que pueda observar la asistencia actual que existe en todas las instalaciones. Así, podrá verse en una tabla las personas que se encuentran en cada una de las instalaciones, en que puestos están trabajando cada uno de los empleados y desde que hora se encuentran en dichas instalaciones. En la vista que se desempeñe esta funcionalidad, deberá aparecer también la hora y fecha actual. Por último, deberá poderse descargarse en formato PDF los datos de esta tabla, y por lo tanto guardar en la base de datos la información de quién, cuándo y que información contiene el informe.

Después, un administrador debe poder conocer las horas que lleva trabajadas cada uno de los empleados, por lo que el usuario deberá poder seleccionar que empleados y que periodo filtrar (los mismos por los que podía filtrar el rol de empleado). Al igual que con la funcionalidad anterior, se podrá generar un informe en formato PDF, cumpliendo con la necesidad de registrar los datos necesarios.

Otra funcionalidad, complementando a la funcionalidad que tiene el usuario de enviar notificación, un administrador deberá poder observar en una tabla todas aquellas notificaciones que se encuentran sin resolver, y por lo tanto en estado “Pendientes”. De aquí, el administrador podrá acceder a una página para resolver la notificación, pudiendo ser necesaria la opción de registrar algún acceso a un empleado si es que éste ha tenido algún problema a la hora de acceder o salir de alguna instalación, o bien resolverla sin realizar ninguna acción.

Al igual que un empleado, un administrador podrá cambiar de rol en cualquier momento, lo que lo deberá llevarlo a ver la página de inicio del rol de empleado. También podrá cerrar sesión, lo que deberá llevarlo a la página de iniciar sesión.

Este primer conjunto de funcionalidades queda reflejado en el próximo caso de uso, quedando muchas funcionalidades más por explicar tras la siguiente imagen.

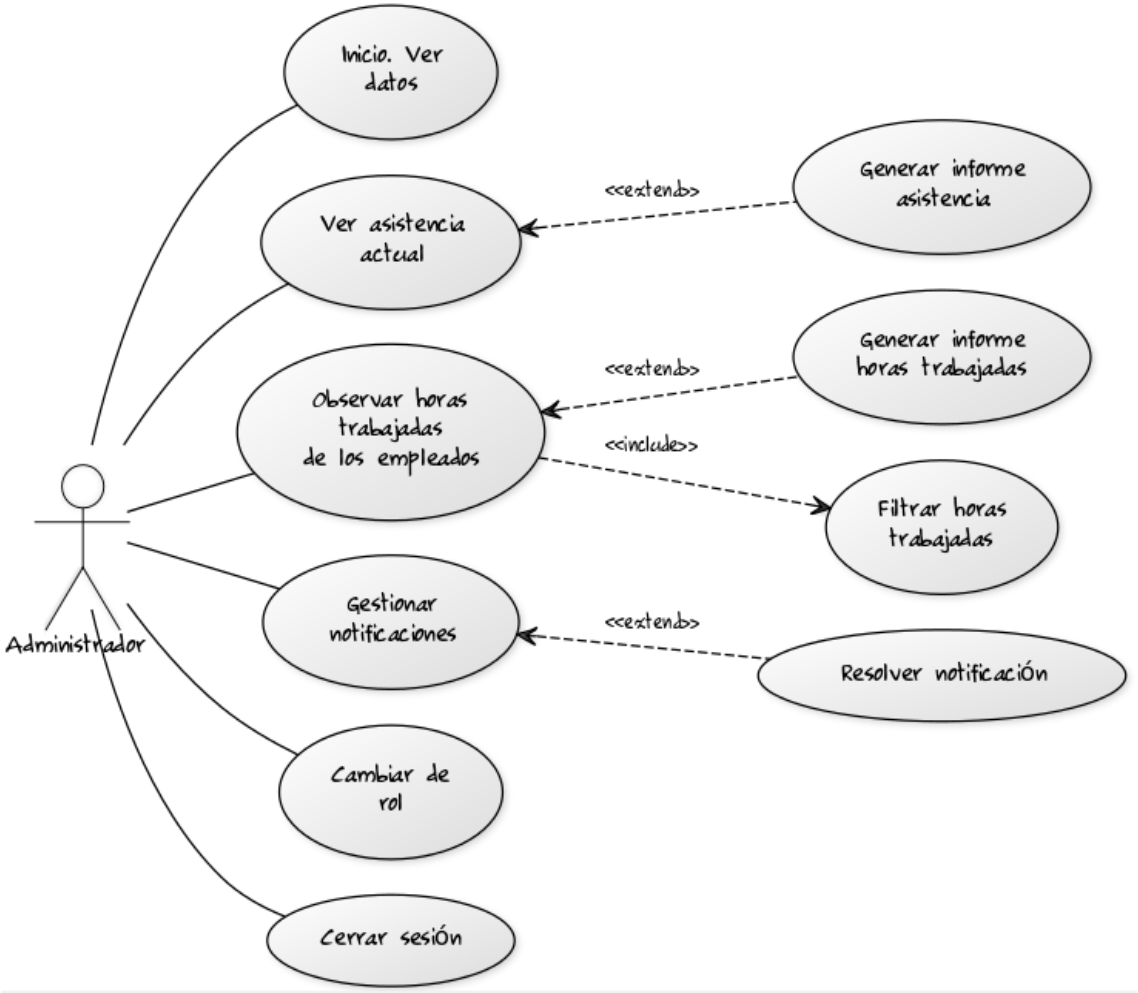


Figura 11: Diagrama de caso de uso. Funcionalidades de administrador 1.

Además de estas funcionalidades básicas, el administrador tendrá la posibilidad de gestionar en el sistema varios conjuntos de datos.

Así, primero de todo el administrador podrá observar en una tabla que usuarios han accedido a la página web a modo de registro de sesiones. Aquí podrá observar quién ha entrado, con qué rol y cuándo ha accedido a la página web. Además, también podrá observar quién, cuándo y sobre qué se han generado diferentes informes.

Además, el administrador podrá dar de alta y baja a usuarios en la página web (podrá crear usuarios con el rol de empleado o administrador). Para esto deberá rellenar un formulario con todos los datos que necesita un empleado. Podrá buscar usuario por DNI, editarlos e incluso borrarlos. Además, se debe poder asignar una tarjeta RFID al usuario creado en el sistema. Así, para facilitar todas estas labores, se deberá poder acceder a una vista donde puedan verse todos los usuarios que existen en el sistema.

Todo esto ocurrirá igual con la gestión de instalaciones. Se deberá poder crear una instalación, buscarla, editarla y borrarla, también deberá poder acceder a una vista en la que puedan conocer todas las instalaciones registradas en el sistema. Estas instalaciones deberán ser las mismas que administrador podrá asignar como principal a cada uno de los usuarios creados.

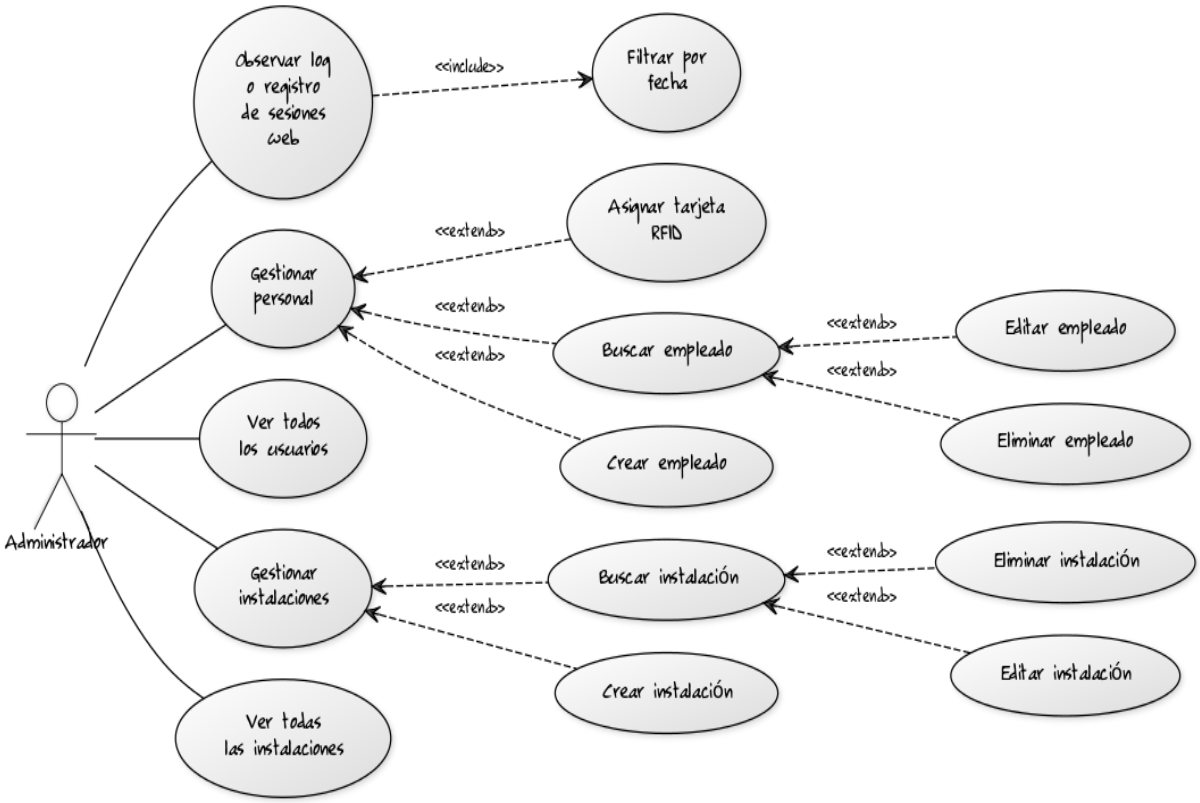


Figura 12: Diagrama de caso de uso. Funcionalidades de administrador 2.

Por último, el administrador también podrá gestionar los horarios. Al igual que con las instalaciones y con los empleados, deberá tener las opciones de crear un horario, buscarlo y borrarlo. También podrá asignar o desasignar alguno de estos horarios a un empleado, de forma que luego éste, en su respectiva vista de la página web, pueda ver rápidamente que horario tiene semanalmente.

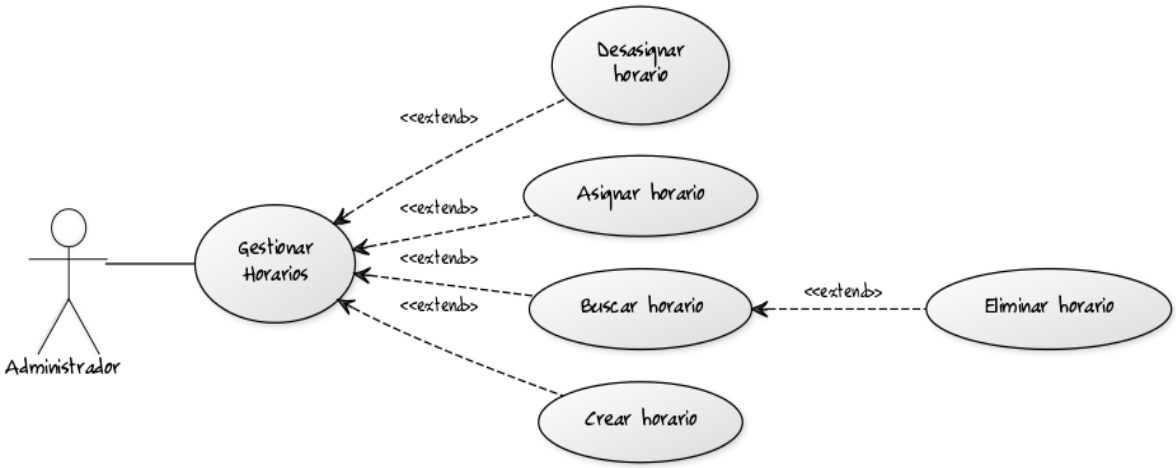


Figura 13: Diagrama de caso de uso. Funcionalidades de administrador 3.

Con todas estas funcionalidades, quedaría especificado el diseño de lo que podrá realizar cada uno de los usuarios en la plataforma web.

Para más exactitud, se muestra a continuación dos diagramas de navegabilidad web, una para empleado y otra para administrador respectivamente.

- Diagrama de navegabilidad para empleado (todas las páginas se tomarán como pestañas, por lo que desde todas se podrá acceder a todas, salvo a recordar contraseña, que deberá ser accedida sólo desde el inicio de sesión):

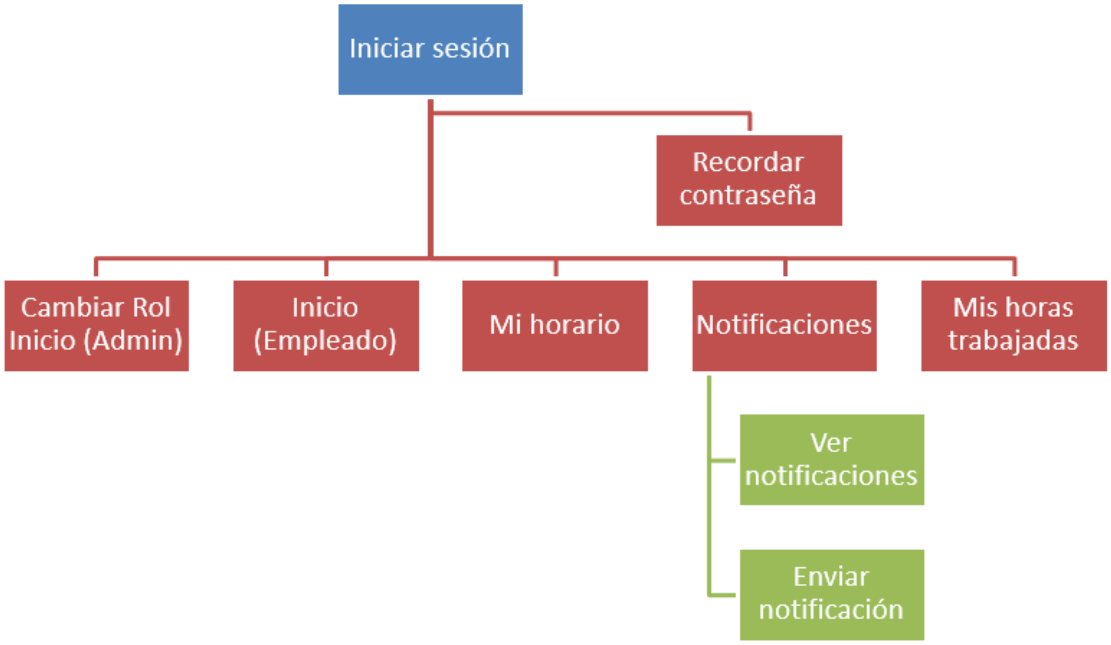


Figura 14: Diagrama de navegabilidad. Empleado.

- Diagrama de navegabilidad para empleado (se toman las mismas consideraciones que con el diagrama anterior):

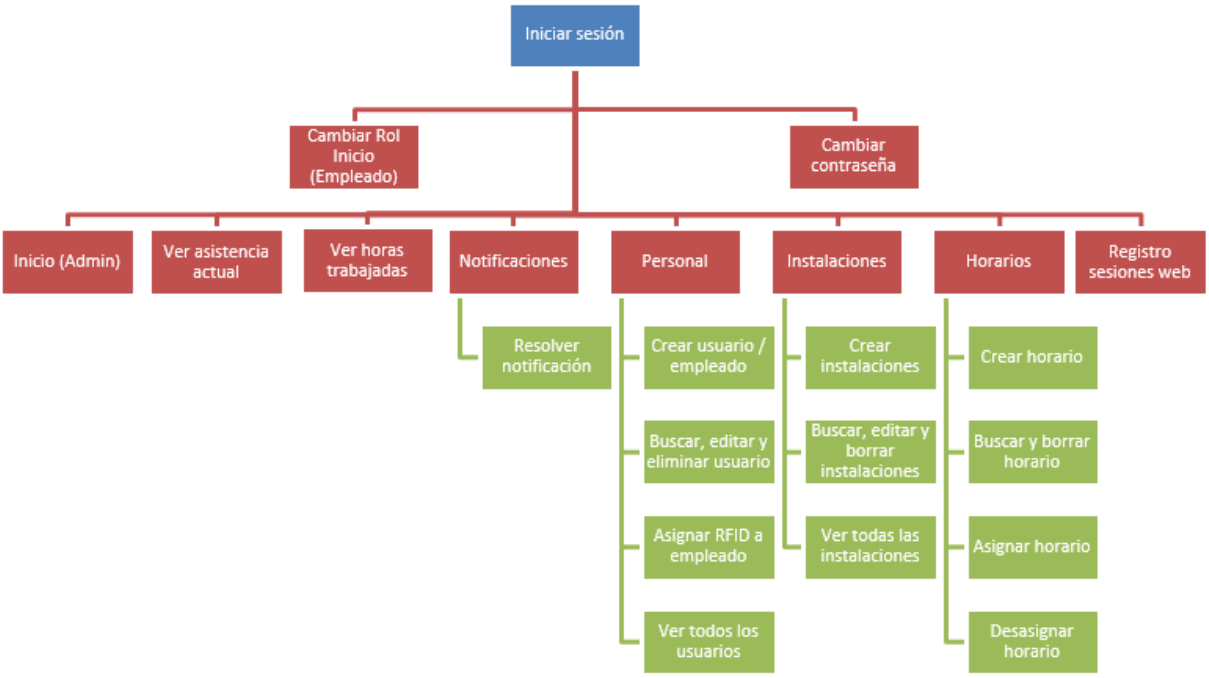


Figura 15: Diagrama de navegabilidad. Administrador.

5.3 COMUNICACIÓN:

La última parte que compone este proyecto vendrá determinada por como deberán comunicarse ambos sistemas (hardware y software).

Para esto se deberá usar el desarrollo de un servicio web para comunicar la base de datos del sistema software con las lecturas de tarjetas RFID que haga el sistema hardware.

Esto se podrá hacer haciendo uso de la respectiva librería para desarrollos en microcontroladores Arduino o similares, que permiten la implementación de un cliente de un servicio web. Así en el servidor donde esté alojada la base de datos, junto con la página web asociada, se podrá crear el correspondiente servidor para dicho servicio web.

6 DESARROLLO

En este capítulo de la memoria se va a describir como se han implementado todas y cada una de las partes que se han explicado en el apartado de la arquitectura que tiene el sistema. Antes de continuar, decir que no se verá el código completo de ninguna de las clases o funcionalidades descritas debido a su extensión.

Al igual que en los apartados anteriores, habrá tres partes diferenciadas:

6.1 HARDWARE:

La implementación del hardware se ha realizado haciendo uso del entorno de desarrollo proporcionado por Arduino. A continuación, se muestra el IDE de Arduino junto con un ejemplo sencillo que permitirá explicar cómo se ha implementado el código:

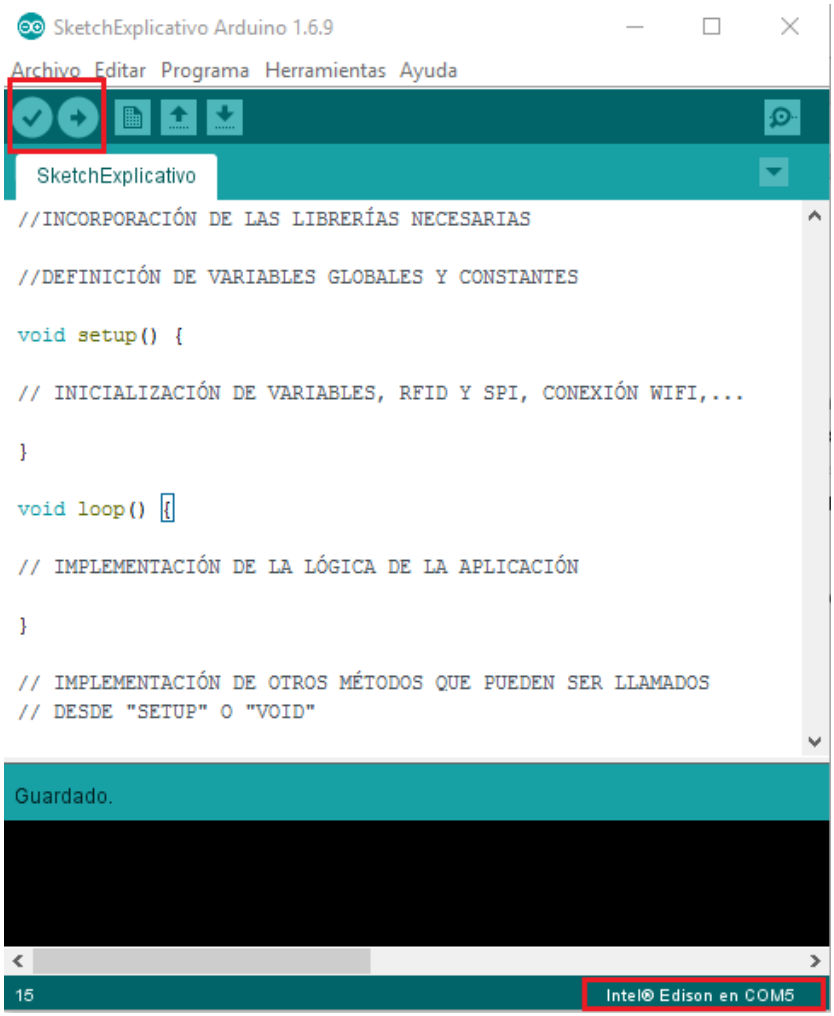


Figura 16: Arduino IDE. Estructura sketch Arduino.

Primero de todo, decir que la programación de Arduino está basada en programación C++, tal y como se adelantó en apartados anteriores.

Tras la anotación anterior, decir que las únicas librerías de Arduino que han sido necesarias son: SPI.h, Wifi.h y RFID.h. Todas estas librerías pueden ser encontradas fácilmente en la página oficial de Arduino.

Después se podrán declarar la variables y constantes que vayan a ser necesarias. En este caso se declararon variables booleanas para poder especificar los estados y constantes usadas para facilitar el trato con los diferentes pines correspondientes a los componentes electrónicos.

En “setup” se inicializan las comunicaciones de RFID y WIFI para poder hacer la lectura de tarjetas y conexión a la red respectivamente.

Por último, en “loop” se establecen toda la lógica para cumplir con la máquina de estados asociada al funcionamiento del sistema hardware, lo cual, para facilitar su implementación se crearon métodos fueran del “loop” que son llamadas cuando son necesarios.

Para terminar con el sistema hardware, aquí se puede ver como el recuadro rojo dibujado abajo nos muestra que se está programando sobre el microcontrolador de Intel Edison. El otro recuadro de arriba nos muestra los botones de compilar el código y subir al microcontrolador respectivamente.

6.2 SOFTWARE:

Para la explicación de cómo se ha implementado el sistema software se expondrán ejemplos de código de cada capa establecida por la arquitectura usada y algunas vistas de la página web.

Para empezar, decir que la implementación de la base de datos se realizó sobre la capa más baja de la arquitectura, capa que está basada en el uso de JPA (Java Persistence API). JPA permite crear cada una de las entidades de la base de datos basándonos en objetos normales de java. A continuación, un ejemplo donde se han señalado con recuadros rojos enumerados los detalles más importantes.

Como se podrá observar, JPA se basa en el uso de anotaciones para especificar como se mapeará la clase java en una entidad de la base de datos.

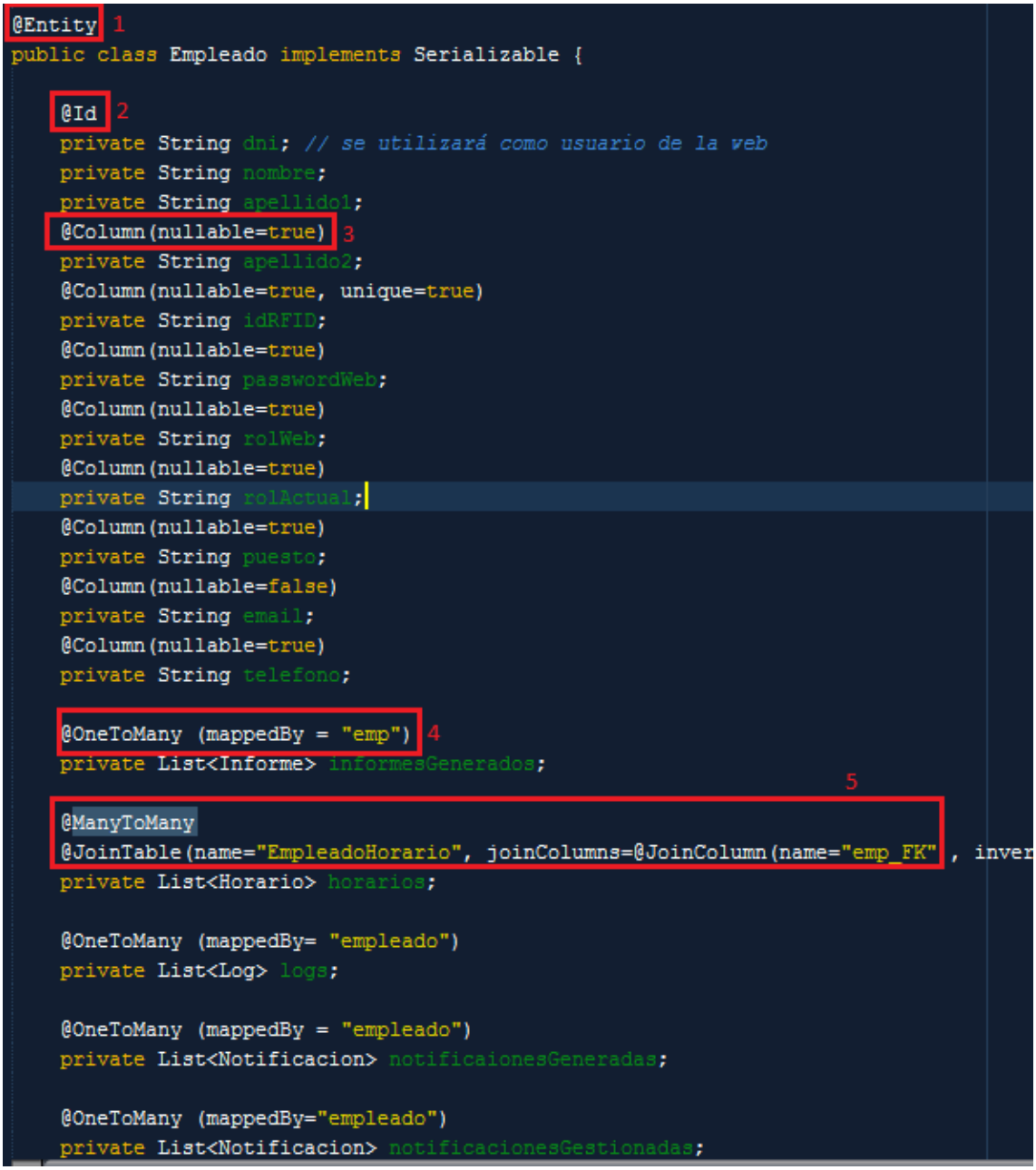


Figura 17: Captura entidad JPA.

En el recuadro número 1 se puede observar la primera anotación (expresadas con el símbolo “@” y el nombre de la anotación) el cual nos especifica que este objeto creado será mapeado como entidad. Los objetos de este tipo deben implementar la interfaz “Serializable” y ser clases de tipo POJO. Esto implica que todo atributo debe tener su respectivo “Get” y “Set”, además el constructor sin argumentos, debe ser una clase de primer nivel y no debe ser “final” ni debe tener miembros de este tipo.

En el recuadro número 2 se puede observar que anotación identifica el atributo que más tarde será mapeado como el identificador de la tabla de la base de datos (en algunas entidades se han establecido algunos parámetros para que

las claves se autogeneren de forma automática cuando se inserten datos en cada tabla).

En el tercer recuadro queda reflejado que anotación hay que usar para especificar que los atributos serán mapeados como columnas. Como se puede observar se pueden añadir parámetros para establecer si el atributo podrá ser nulo o no y si podrá ser único, entre otros parámetros más como el nombre que tendrá el atributo en la base de datos, la longitud máxima que podrá tener, etc.

En los puntos cuatro y cinco se muestran las anotaciones necesarias para establecer relaciones entre las entidades. Las diferentes anotaciones son @OneToOne, @OneToMany, @ManyToOne y @ManyToMany. Así para crearlas debe crearse una instancia del objeto “entity” que se quiera relacionar. Señalar que en las relaciones @ManyToMany se genera una tabla intermedia, la cual puede cambiarse el nombre y el nombre de los atributos que formarán parte de ésta haciendo uso de @JoinTable.

Esto junto todos los “Getters” y “Setters” de cada atributo, el constructor sin argumentos y los métodos “toString”, “equals” y “hashCode” para poder mostrar como cadena de caracteres y poder comparar objetos.

Esto, junto al fichero de configuración llamado “persistence.xml”, permitirán que estas clases sean mapeadas en el servidos GlassFish como entidades de la base de datos.

Tras esto la siguiente capa, la encargada de establecer la comunicación entre la base de datos y el resto de la aplicación, lo que sería conocido como la capa de negocio, ha sido implementada haciendo uso de EJBs (Enterprise Java Beans).

Al igual que para JPA, a continuación, se muestra una imagen de una clase EJB con diferentes recuadros en rojo señalando algunos de los elementos más importantes que contienen este tipo de clases.

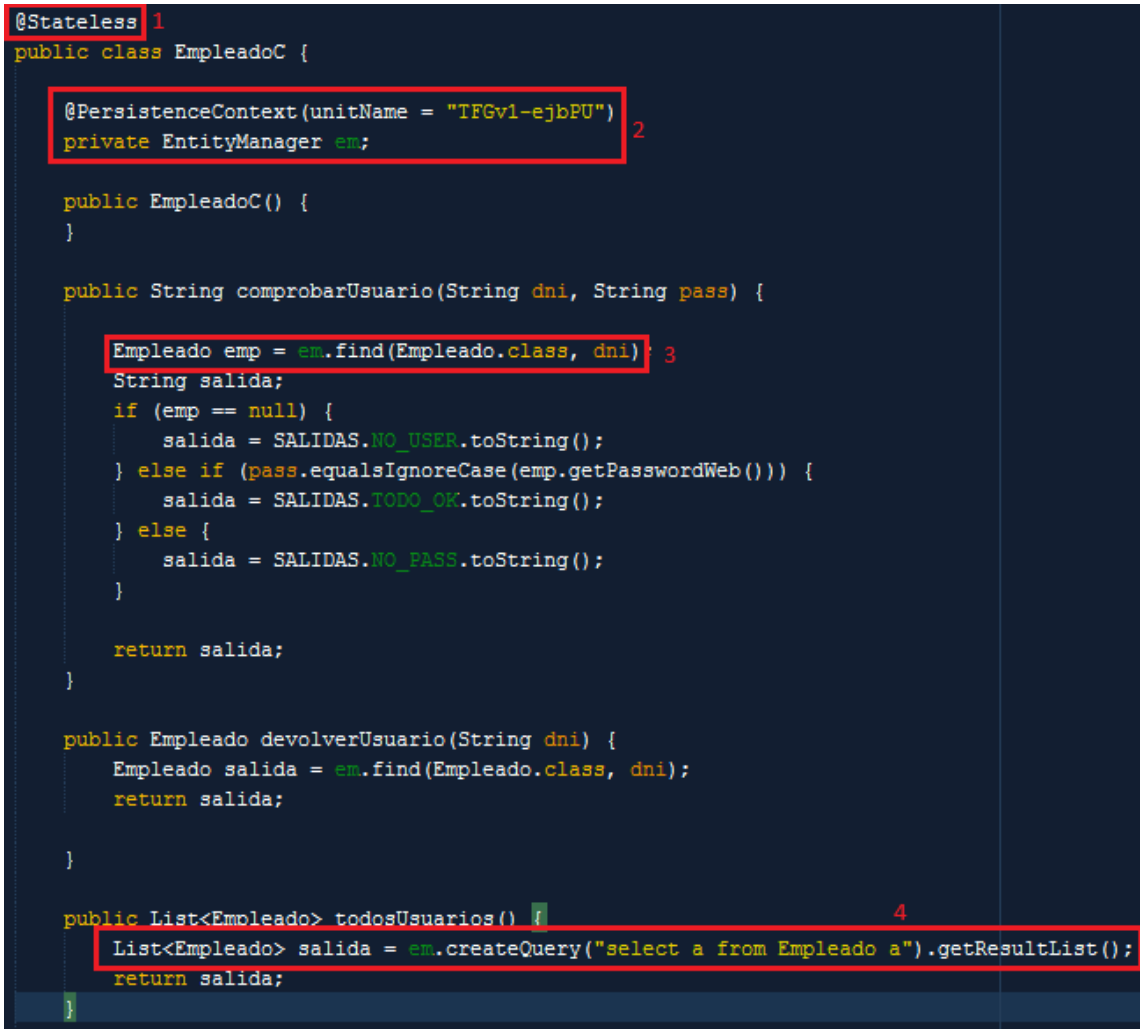


Figura 18: Captura clase EJB.

Así, el primer recuadro se muestra la anotación que indica de que tipo puede ser el EJB. En este caso @Stateless nos indica que el EJB no tendrá estado, es decir, no guardará información después de que sea llamado alguno de sus métodos. Todos los EJB de este proyecto se han declarado de este tipo. Existen dos tipos más, aunque no han sido usados en este proyecto: @Stateful y @Singleton.

En el segundo recuadro se muestra la anotación necesaria (@PersistenceContext) para especificar al EJB con qué unidad de persistencia debe comunicarse y donde se establece el objeto Java que va a ser necesario para realizar la comunicación, EntityManager.

Con este objeto se podrán hacer consultas a la base de datos haciendo uso de algunos métodos preestablecidos que permiten hacer consultas sencillas como en el caso de la tercera imagen, lo cual nos devuelve el empleado que tenga el DNI que se le pasa como argumento.

También se han creado consultas más complejas como puede observarse en el cuarto recuadro. Para esto se ha utilizado un lenguaje de base de datos denominada JPQL (Java Persistece Query Language)

Haciendo uso de estos elementos, se han creado todos y cada uno de los métodos necesarios para crear, leer, actualizar o borrar información de la base de datos. Los métodos que ofrece EntityManager para estas acciones son persist(), find(), merge() o remove() respectivamente, aunque si se quieren hacer consultas o acciones más complejas es necesario el uso de JPQL.

Por último, para la llamada de cada uno de los métodos de esta capa es necesario incluir en cada clase que vaya a hacer uso de sus métodos las siguientes líneas:

```
@EJB
private EmpleadoC empC;
```

Figura 19: Incluyendo EJB en clase JSF.

Estas líneas ya son incluidas en lo que forma parte de la siguiente capa de la arquitectura. Esta capa ha sido implementada haciendo uso de JSF (Java Server Faces). Esta tecnología permite crear páginas web dinámicas de forma que la información pasada desde el servidor (desde las capas anteriores) puedan ser mostradas en la web.

JSF necesita de un lenguaje complementario para ser usado, XHTML. Esto junto HTML, CSS y BootsTrap han permitido la creación de la página web.

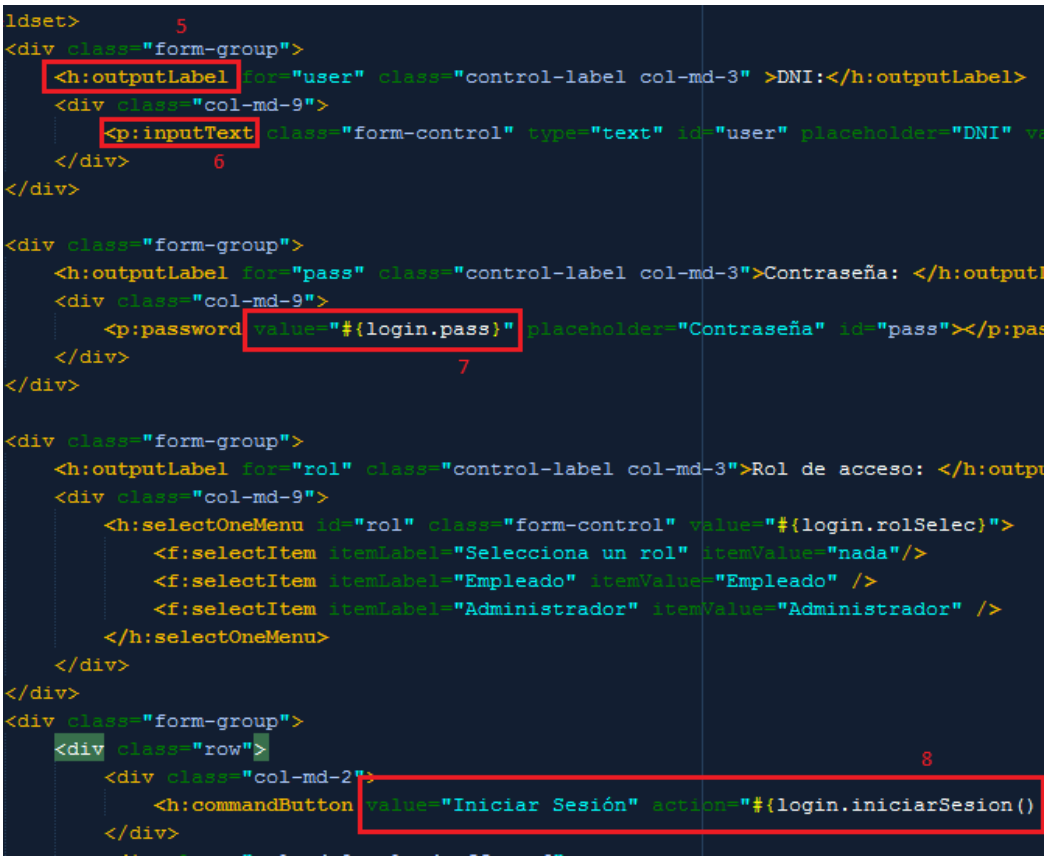
A continuación, se muestran varias imágenes de ejemplos sobre cómo se han implementado los componentes de esta capa y donde se han señalado los elementos más importantes o destacados.

```
@Named 1
@SessionScoped 2
public class SessionBean implements Serializable { 3

    @EJB
    private EmpleadoC empleadoC;
    @EJB
    private NotificacionC notC; 4

    //<editor-fold defaultstate="collapse" desc="VARIABLES">
    private Empleado empLogueado;
    private String rolActual;
    private List<Notificacion> notificaciones;
    private Notificacion notParaResolver;
    //</editor-fold>
```

Figura 20: Captura JSF 1.



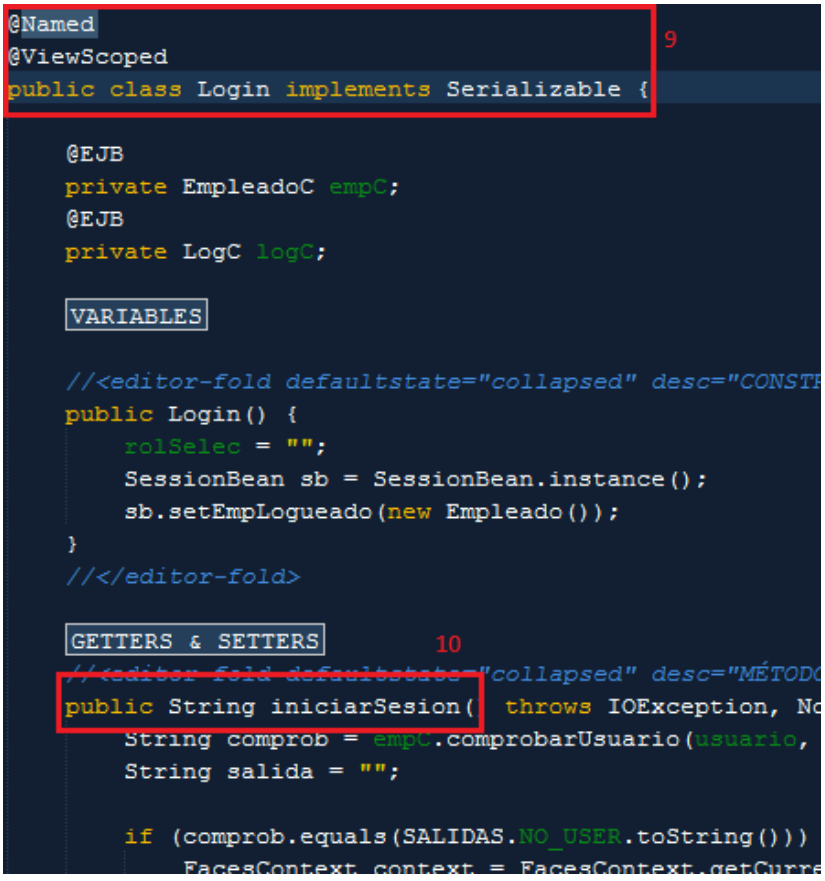


Figura 20 y 21: Captura XHTML y JSF 2

En el primer recuadro se muestra la anotación @Named la cual permite que la clase JSF sea llamada desde una página XHTML.

En el segundo recuadro se muestra la anotación que indica la vida que tendrá el objeto que se cree de esta clase. En este proyecto se han usado @SessionScoped para guardar los datos de cada sesión de un usuario y @ViewScoped para guardar los datos necesarios de cada una de las vistas de la página web.

En el tercer recuadro se muestra como todas las clases JSF deben implementar la interfaz Serializable

En el cuarto recuadro aparecen dos EJBs que se han incluido en este JSF para poder rescatar información de la base de datos haciendo uso de los EJBs que ahí se exponen.

En el quinto recuadro se muestra un componente de XHTML usado para mostrar una etiqueta.

En el sexto recuadro se muestra como se ha usado un componente del framework PrimeFaces, cuyo uso es similar al de un componente XHTML.

En el séptimo recuadro se ve como se pasan valores desde una página web a un JSF. Para esto es necesario que en la clase Login todos los atributos que vayan a ser usados por una página web tengan asociados un Get() y un Set(). Notar que para llamar a un JSF es necesario usar “#{…..}”.

En el octavo recuadro se ve cómo puede dársele una acción a un botón llamando a un método de JSF. El método al que se está llamando se muestra en el recuadro décimo y su funcionalidad tratará de mostrar un mensaje de error en caso de que haya ocurrido algo como que la contraseña es errónea o el DNI no está registrado en el sistema o si todo está correcto permitirá que el usuario inicie sesión y acceda a la primera página de inicio de alguno de los roles permitidos.

En el noveno recuadro se muestra una nueva clase JSF en el que se ha usado un JSF de tipo ViewScoped para guardar solo información durante el tiempo que el usuario se encuentra en esta vista de la página web.

Además de estos componentes, decir que también se ha hecho uso de la tecnología AJAX (Asynchonous JavaScript And XML). Ésta permite actualizar algún componente de la vista web sin necesidad de refrescar la vista entera basándose en los eventos que puedan ocasionarse.

De esta forma, todo el software se ha ido implementando según las especificaciones y siguiendo este estilo de desarrollo.

6.3 COMUNICACIÓN:

Para la implementación de la comunicación entre los sistemas hardware y software se ha usado un servicio web REST (Representational State Transfer). Para esto se ha necesitado crear en el servidor un servidor de peticiones REST que capturase la información que fuese enviada desde el cliente (sistema hardware).

A continuación, se muestra una imagen del servicio web implementado en el servidor y otra imagen del servicio web implementado en el cliente:

```java
@Stateless
@Path("accesos")
public class AccesoFacadeREST extends AbstractFacade<Acceso> {

    @PersistenceContext(unitName = "TFGv1-warPU")
    private EntityManager em;

    public AccesoFacadeREST() {
        super(Acceso.class);
    }

    @PUT
    @Override
    @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    public void create(Acceso entity) {
        super.create(entity);
    }

    @POST
    @Path("{tarjeta}/{instalacion}/{accion}")
    @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    public String edit(@PathParam("tarjeta") String tarj, @PathParam("instalacion")
        String salida;
```

```java
if(conectado){
  // Make a HTTP request:
  client.println("POST /TFGv1-war/webresources/accesos/" + tarjeta +"/" + ID_INSTALACION + "/" + valorAccion +" HTTP/1.1");
  client.println("Host: 192.168.0.101");
  //client.println("Connection: close");
  client.println();
```

Figura 22 y 23: Servidor y cliente servicio web.

Como se puede comprobar, el cliente hace una petición al servidor mandando los parámetros necesarios que luego serán leídos en el servidor para hacer con ellos lo que sea necesario y devolverle una respuesta al cliente para que éste pueda continuar con su siguiente estado.

7 PRUEBAS

Para la realización de pruebas del sistema se ha hecho uso de Selenium IDE. Para hacer uso de esta aplicación es necesario añadir al navegador Firefox esta herramienta como extensión. Este sistema está basado en la comprobación de asertos que se establecen con antelación al testeo que se realiza automáticamente con datos que se le también se pueden preestablecer antes de iniciar toda la prueba.

Así, se han realizado pruebas de la página web de cada camino sin ciclo en el diagrama de navegabilidad para ir comprobando su correcto funcionamiento.

Tras haber analizado cada uno de los caminos, se ha comprobado que al integrar en el sistema cada uno de estos, todos aquellos que se encontraran relacionados por la información que gestionaban tuvieran una información consistente.

También se han realizado pruebas de acceso concurrente, es decir, que varios usuarios puedan acceder a la aplicación al mismo tiempo sin impedir el buen funcionamiento. Esto se ha realizado para comprobar que todas las acciones que realicen diferentes usuarios queden reflejadas para el resto en caso de estar relacionadas.

Tras estas pruebas del sistema software, se han probado todos los estados de la máquina de estados que componían el sistema hardware. Para esto se ha obligado al sistema a pasar por todos estos estados desconectado la conexión a internet o simplemente esperando el tiempo necesario para que el sistema piense que el botón se ha pulsado por accidente o simplemente porque ha ocurrido algún error.

Además, durante la implementación del sistema se hicieron pruebas del buen funcionamiento de todos los componentes de forma atómica. Después de que todo funcionara correctamente se hicieron pruebas de integración de los componentes para comprobar que la comunicación de información entre un componente y otro se hacen correctamente. Por último, también se hicieron pruebas para comprobar que todo el sistema funcionaba correctamente.

Con todo esto, el sistema ha quedado probado. Esto, junto al buen funcionamiento que se le puede observar, hace pensar que la aplicación no tenga ningún problema grave en lo que se refiere a implementación por lo que podría hacerse uso de este sistema sólo realizando algunos ajustes.

8 LÍNEAS FUTURAS

Debido a que este proyecto está dirigido a cumplir con lo estipulado en las normas de un Trabajo de Fin de Grado, muchas de las ideas que se pensaron antes de iniciarlo se tuvieron que dejar a un lado debido a la gran complejidad que tendría el sistema a desarrollar.

Este proyecto podría ampliarse añadiendo, por ejemplo, la función de poder controlar también las horas extras que un empleado ha dedicado o gestionar las horas que un empleado pudiera trabajar en una empresa, pero sin asistir a las instalaciones de ésta. Algunos de estos casos pueden ser el de comerciales que deben desplazarse en la calle contabilizándose estas horas como horas trabajadas o por ejemplo algún empleado que pueda trabajar con escritorio remoto desde su casa).

Podría ser un componente más de un ERP (Enterprise Resource Planning) en el módulo de Recursos Humanos facilitando así la posterior generación de las nóminas en algunos trabajos en los que ésta dependa sustancialmente de las horas que un empleado haya trabajado.

9 CONCLUSIONES

Tras todo esto, decir que este proyecto se ha solucionado completamente según los objetivos que se marcaron al inicio. Para alcanzar esta solución he tomado varias decisiones que desde el principio de todo ha sido bastante satisfactorias. Sin embargo, también tomé algunas decisiones que tuve que cambiar con el paso del tiempo.

Estas malas decisiones me provocaron un pequeño retraso en la etapa de desarrollo del sistema hardware. Todo fue debido a la decisión sobre que microcontrolador usar para la implementación de todo el sistema hardware.

La primera idea fue la de utilizar Arduino en su versión UNO, pero debido a la necesidad de tantos componentes externos venía la problemática de como conectar la totalidad de pines que necesitaban cada componente. Esto se produjo porque el módulo RFID y el módulo de conexión inalámbrica necesitaban uso la comunicación SPI que posee Arduino. Esto era algo que se podría solucionar con programación, pero para facilitar las cosas se decidió el uso de un microcontrolador que ya contuviera alguno de estos módulos en su placa de expansión, este era el caso de Intel Edison.

Esto me provocó unos pocos días de retraso que me lastraron hasta el final del proyecto provocando que algunos plazos marcados por mí mismo no se cumplieran.

Sin embargo, también valoro la capacidad de reacción que se debe tener para solucionar problemas con el ocurrido. Fui capaz de, en apenas dos o tres días, encontrar una solución fácil y rápida a todo este problema (se valoró la adquisición de una Arduino en su versión MEGA, pero lo descarté debido al precio).

En el caso de los aciertos, valoro en gran medida el haberme marcado los plazos expuestos en el apartado de planificación y por lo tanto de haber dividido el proyecto en todas esas etapas.

El análisis de requisitos y el diseño de la aplicación fueron esenciales para un rápido desarrollo de todo el sistema, ya que, con la salvedad del problema ocurrido con Arduino UNO, es mucho más sencillo implementar un sistema totalmente especificado que alguno que contiene requisitos difusos.

Con todo esto, decir que se han aplicado muchos conocimientos adquiridos en el conjunto de asignaturas que se ofrecen en esta carrera, como son el caso de los conocimientos adquiridos sobre desarrollos en Arduino, desarrollos de páginas web haciendo uso de Java EE y toda la arquitectura que esta tecnología o lenguaje ofrece, conocimientos sobre como diseñar una base

de datos intentando que toda la información que se almacenara fuera consistente y sin problemas de redundancia de datos y por último también se ha hecho uso de los conocimientos de ingeniería del software para el análisis de requisitos y diseño de toda la aplicación. A todo esto, hay que unir el uso de la mentalidad adquirida de como ver los problemas grandes como subproblemas más pequeños para resolverlos de una forma más sencilla.

Así se ha creado un sistema desde cero siguiendo todos los pasos normales para la puesta en marcha del proyecto, documentando cada etapa con anotaciones e imágenes que más tarde han servido para la redacción de esta memoria a modo de documentación del proyecto creado.

La solución que se ha creado cumple los objetivos planteados:

- Analizar, diseñar e implementar un sistema hardware para la lectura de tarjetas con tecnología RFID -> Realizado con éxito, basado en el microcontrolador Intel Edison y la máquina de estados especificada.

- Analizar, diseñar e implementar una plataforma web donde se permita la visualización de la información recopilada por el sistema hardware. Este sistema a su vez planteaba diferentes subobjetivos:

    - Seguimiento de las horas trabajadas por los empleados. -> Implementado como una pestaña a la que pueden acceder tanto empleados como administrador.

    - Seguimiento de estancia actual en las instalaciones. -> Implementado como una pestaña de la página web en el rol del administrador.

    - Posibilidad de generación de informes. -> Implementado en varias pestañas tanto para el empleado como para el administrador donde se ha considerado que puede existir información importante de obtener en formato PDF.

- Analizar, diseñar e implementar un sistema que se encargue de la correcta lectura de las tarjetas con tecnología RFID, así como la gestión de usuarios registrados en el sistema con una tarjeta. -> Realizado entre el sistema hardware para la lectura de las tarjetas RFID y en el software para gestionar que usuarios (junto con gestión de instalaciones y horarios) tienen acceso a la plataforma web y cuales están identificados por una tarjeta RFID.

- Analizar, diseñar e implementar un sistema de comunicación entre el sistema hardware y la plataforma web para poder transmitir los datos de un sistema a otro. -> Implementado mediante el uso de servicios web de

tipo RESTful utilizando como cliente el sistema hardware que manda a través del módulo de conexión inalámbrica las peticiones REST al servidor REST establecido en el sistema software, alojado en el servidor.

Juntos a estos objetivos, también se ha resuelto el objetivo implícito de la gestión de toda la información con la implementación de una base de datos alojada en el servidor del sistema software.

10 REFERENCIAS BIBLIOGRÁFICAS

[1] Java EE 7: https://docs.oracle.com/javaee/7/tutorial/

[2] Arquitectura y diseño de aplicaciones Java EE: http://es.slideshare.net/cptanalatriste/arquitectura-y-diseo-de-aplicaciones-java-ee.

[3] HTML, XHTML and CSS: HTML, HXTML, and CSS bible / Steven M. Schafer. Indianapolis, Ind. : Wiley Pub., 2010.

[4] RESTful: https://docs.oracle.com/javaee/7/tutorial/jaxrs.htm

[5] Arduino IDE: https://www.arduino.cc/en/Main/Software

[6] Intel Edison: http://www.intel.la/content/www/xl/es/do-it-yourself/edison.html

[7] RFID standards – ISO: www.iso.org/iso/livelinkgetfile-isocs?nodeId=15545715

[8] PrimeFaces: http://www.primefaces.org/

[9] Bootstrap: http://getbootstrap.com/

[10] Netbeans IDE: https://netbeans.org/

[11] AJAX: http://www.w3schools.com/ajax/

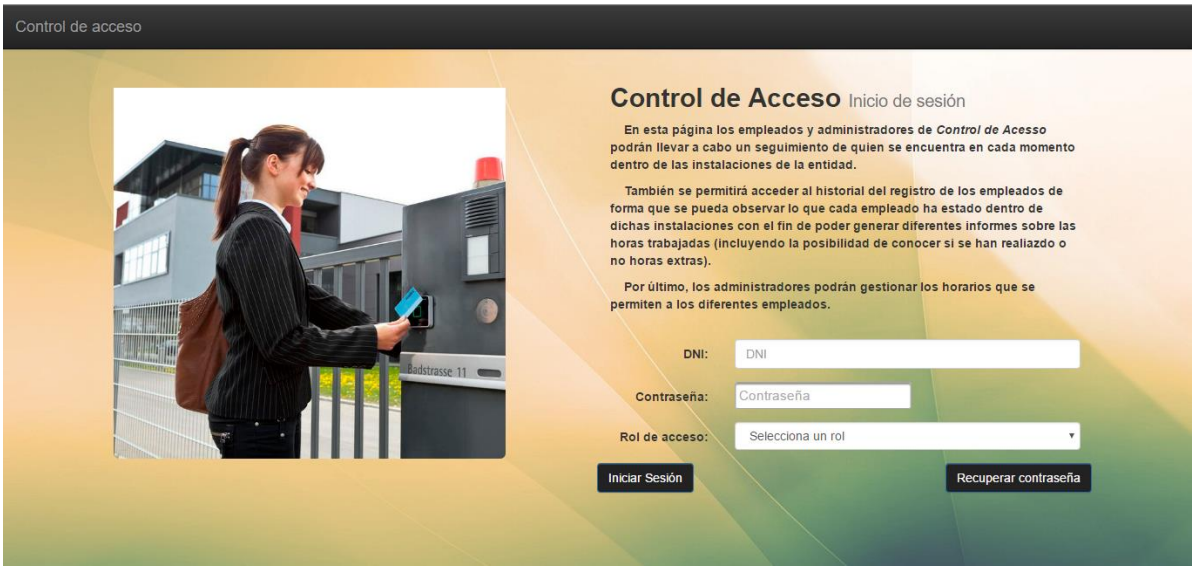
[12] Comunicación SPI: http://www.prometec.net/bus-spi/

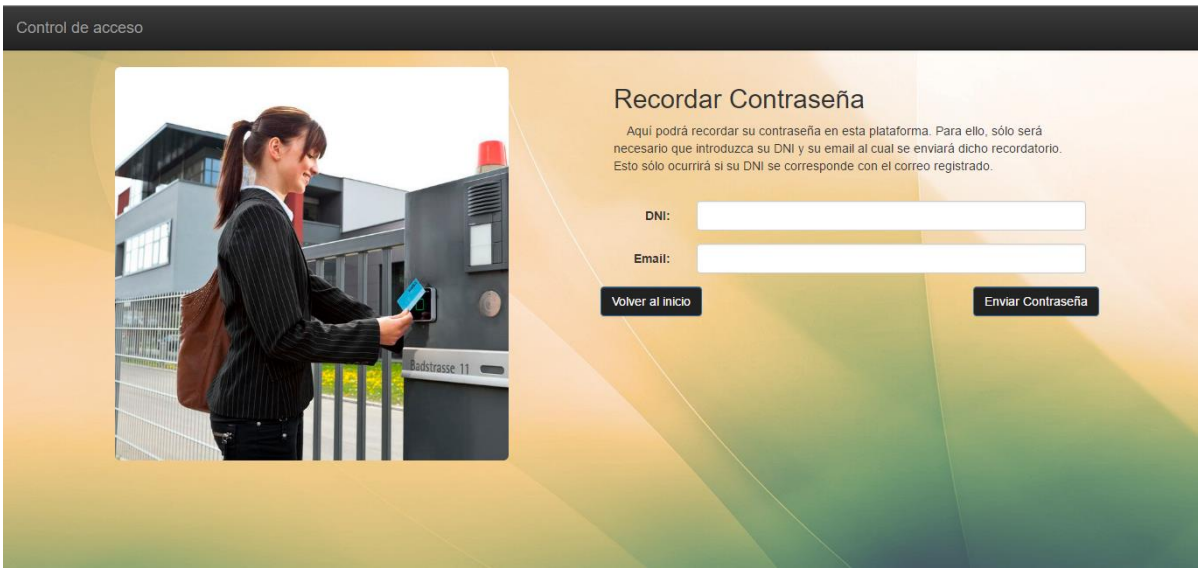
11 ANEXOS

11.1 CAPTURAS DE LA PÁGINA WEB:

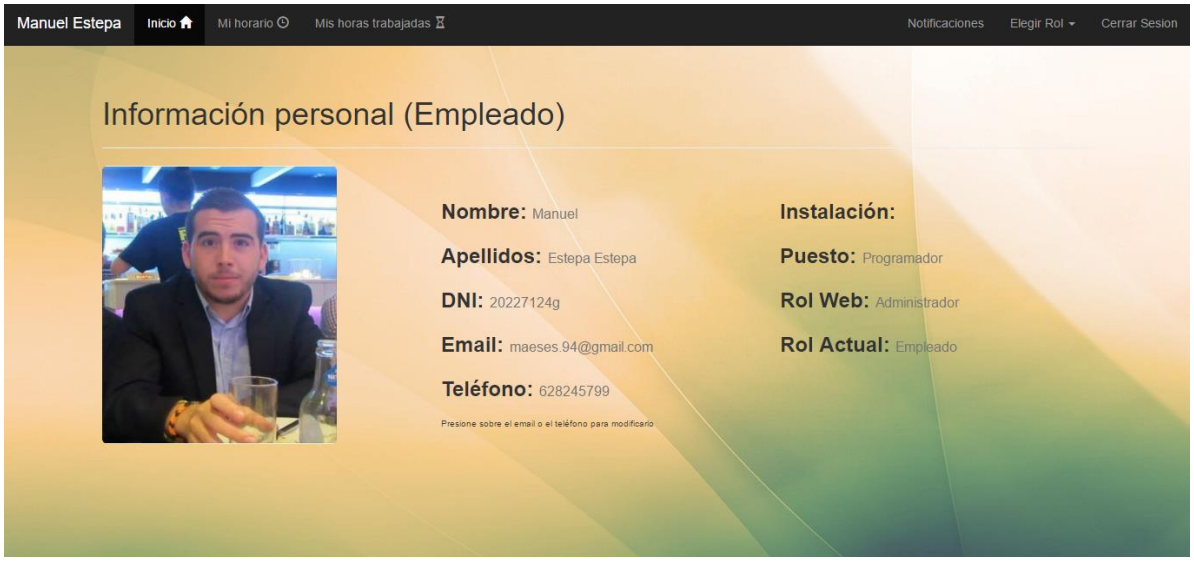
A continuación, se muestran todas las vistas a las que podrán acceder cada uno de los usuarios de la página web, ya sea con el rol de empleado o administrados

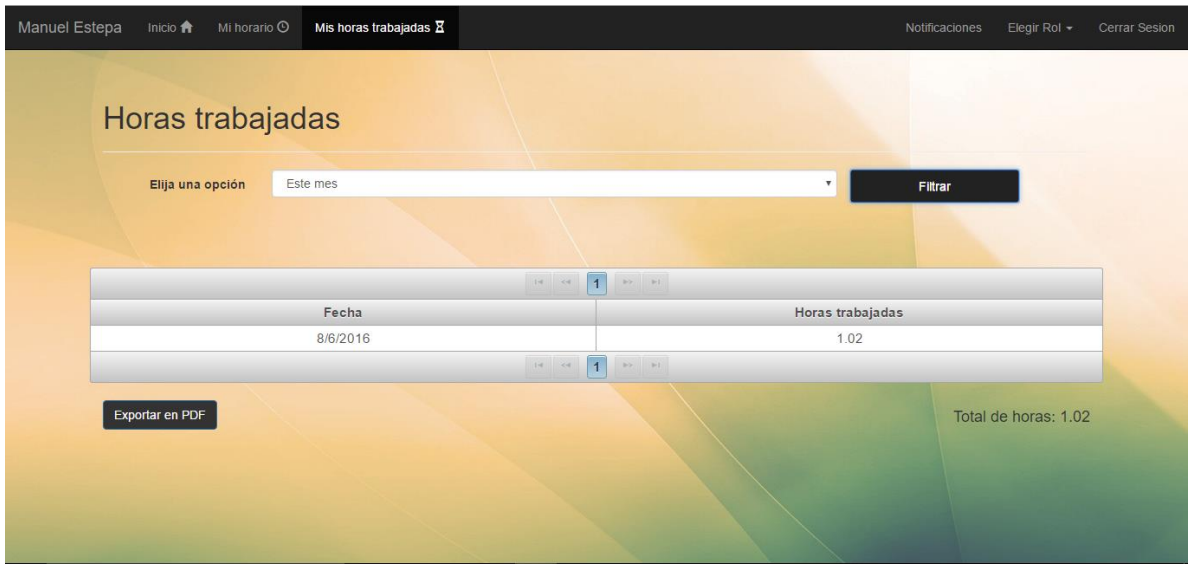
11.1.1 Páginas comunes:

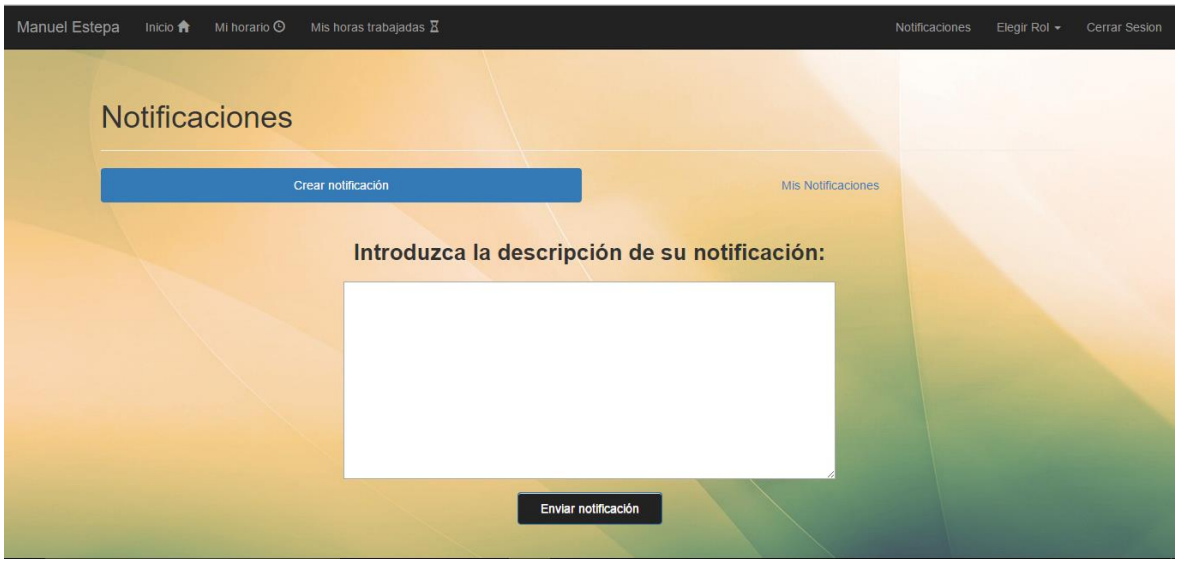


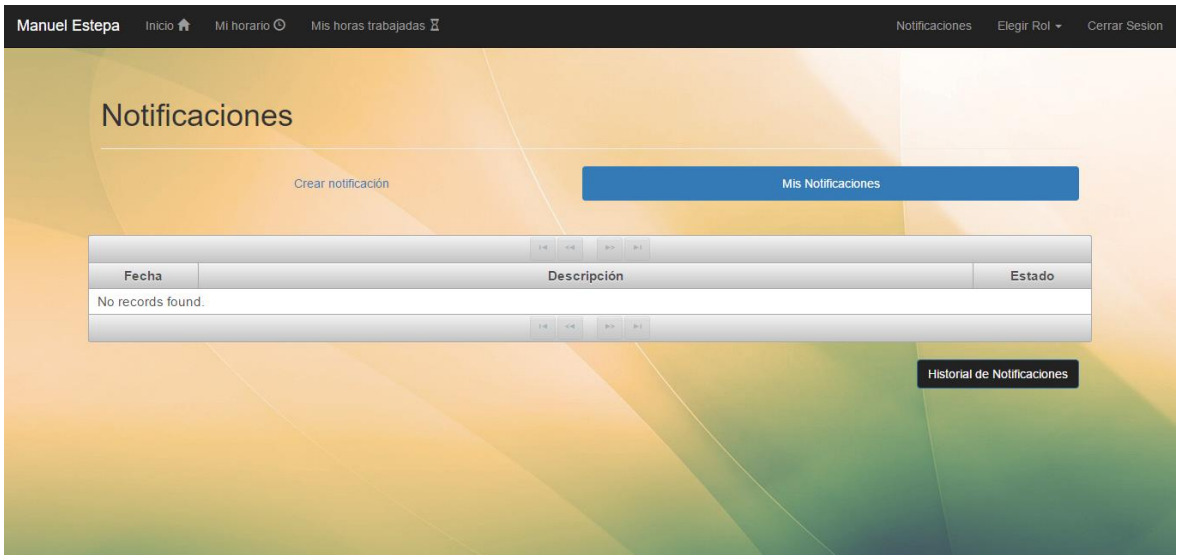


Rol de empleado:

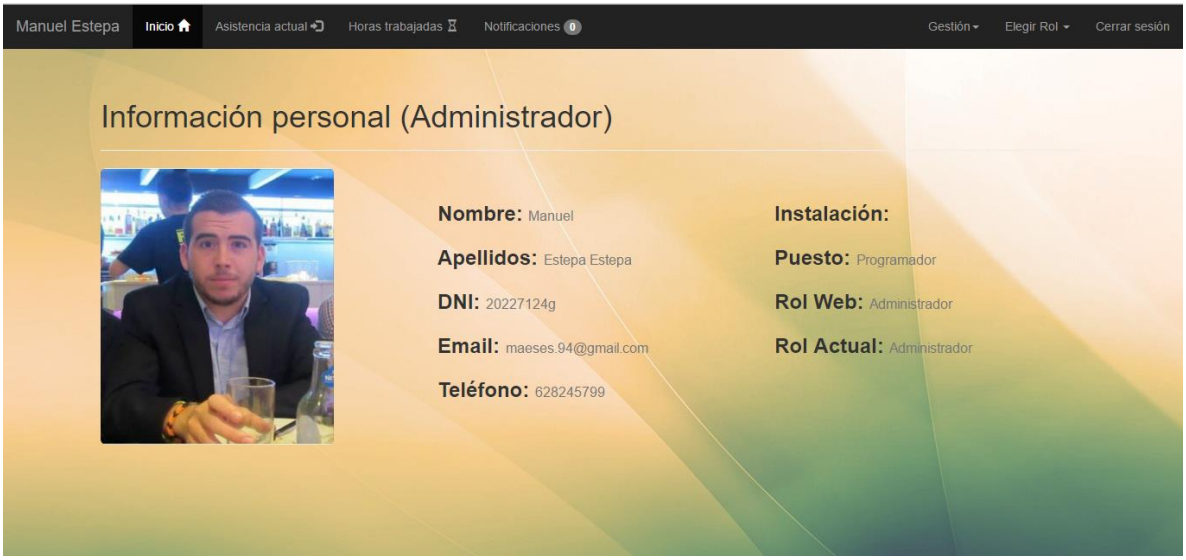


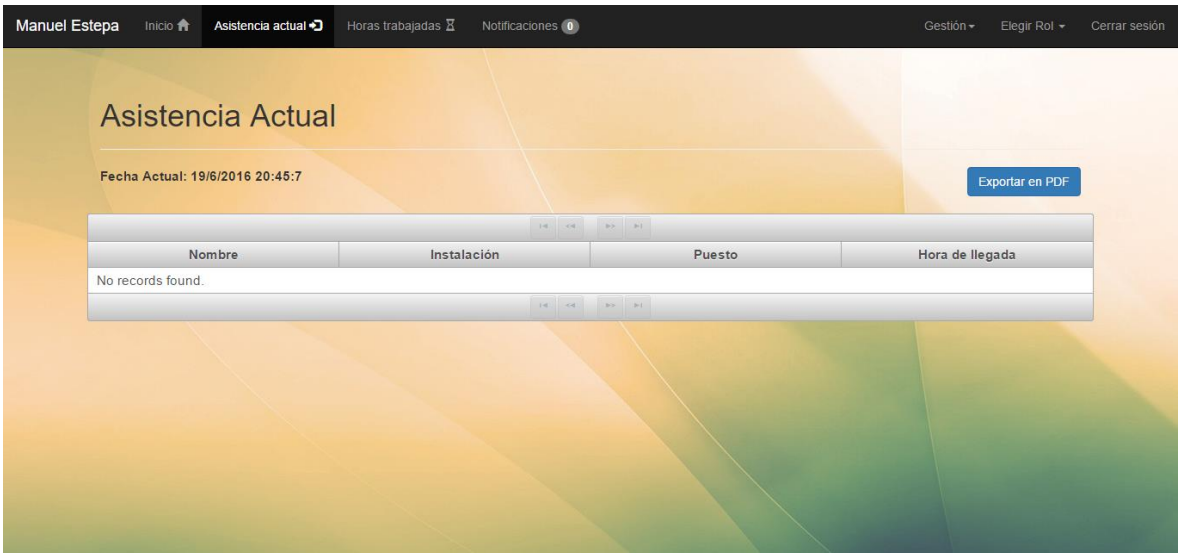


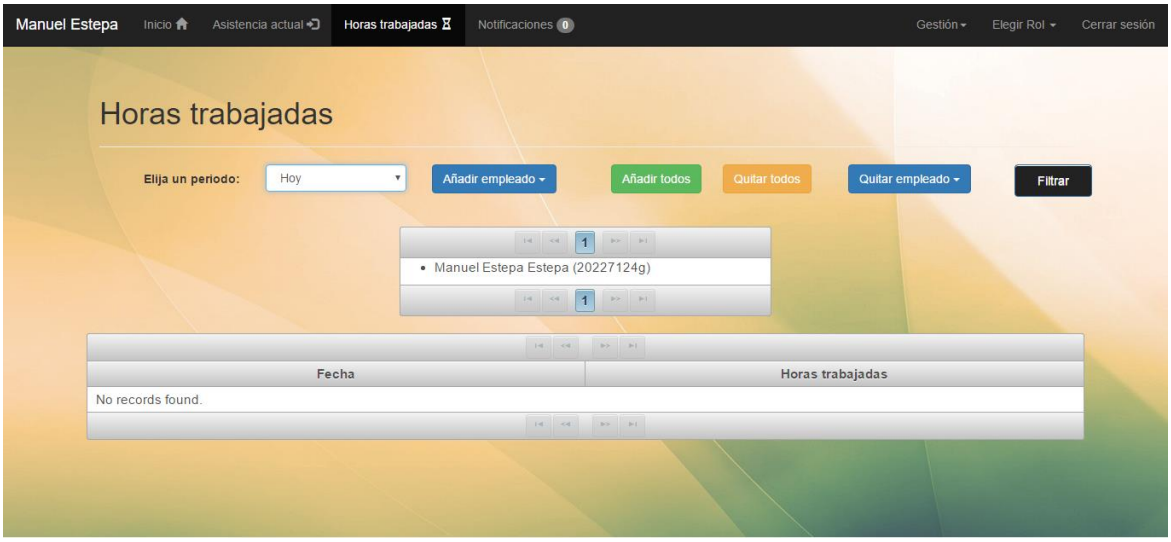

Manuel Estepa
Inicio
Mi horario
Mis horas trabajadas
Notificaciones
Elegir Rol
Cerrar Sesion
Notificaciones
Crear notificación
Mis Notificaciones
Introduzca la descripción de su notificación:
Enviar notificación

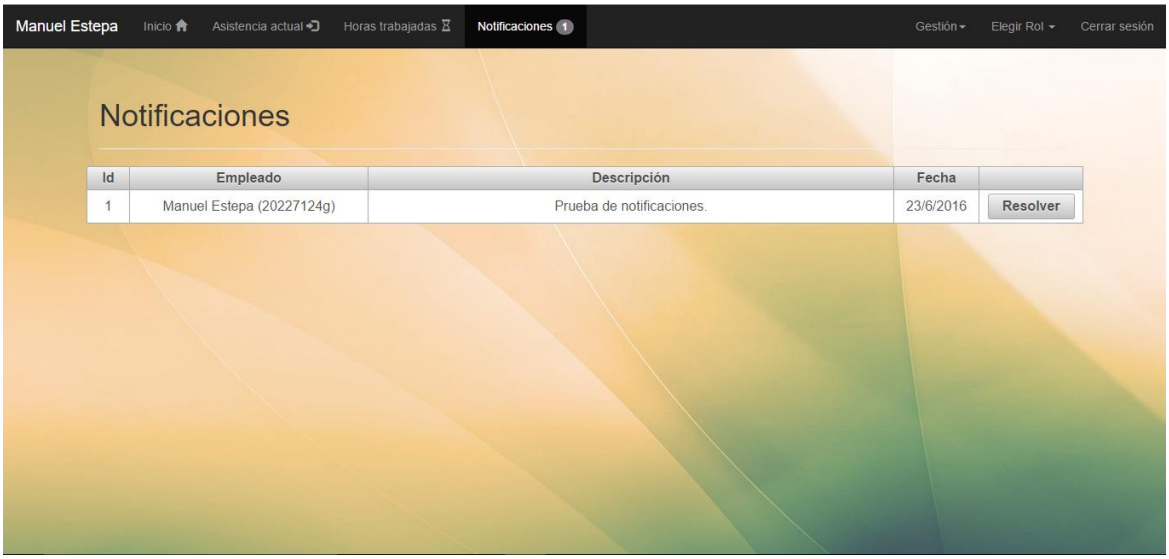

Manuel Estepa
Inicio
Mi horario
Mis horas trabajadas
Notificaciones
Elegir Rol
Cerrar Sesion
Notificaciones
Crear notificación
Mis Notificaciones
Fecha
Descripción
Estado
No records found.
Historial de Notificaciones

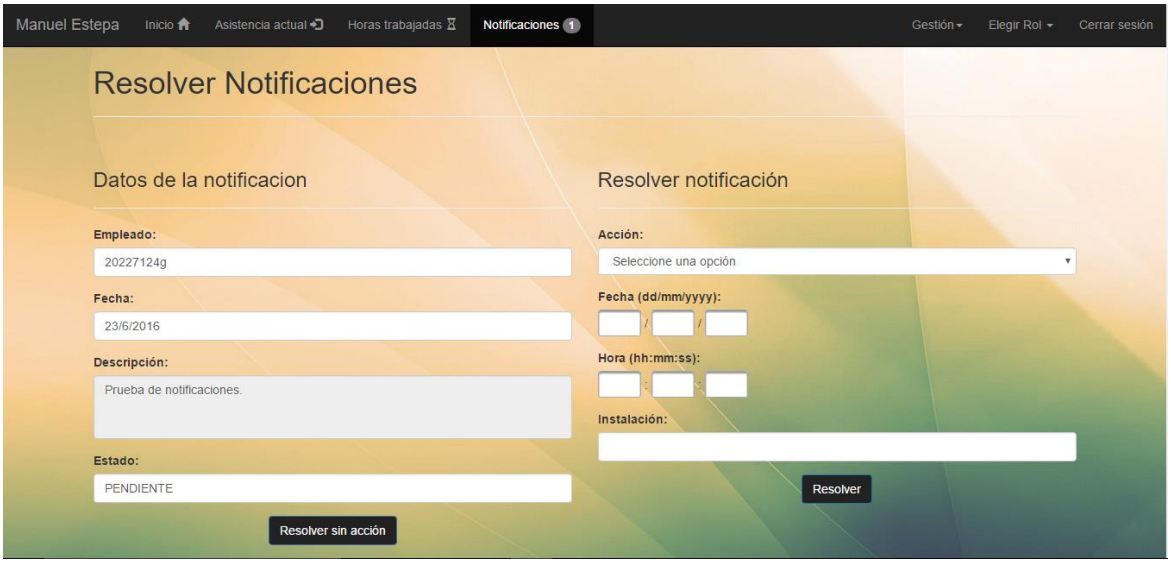
Rol de administrador:

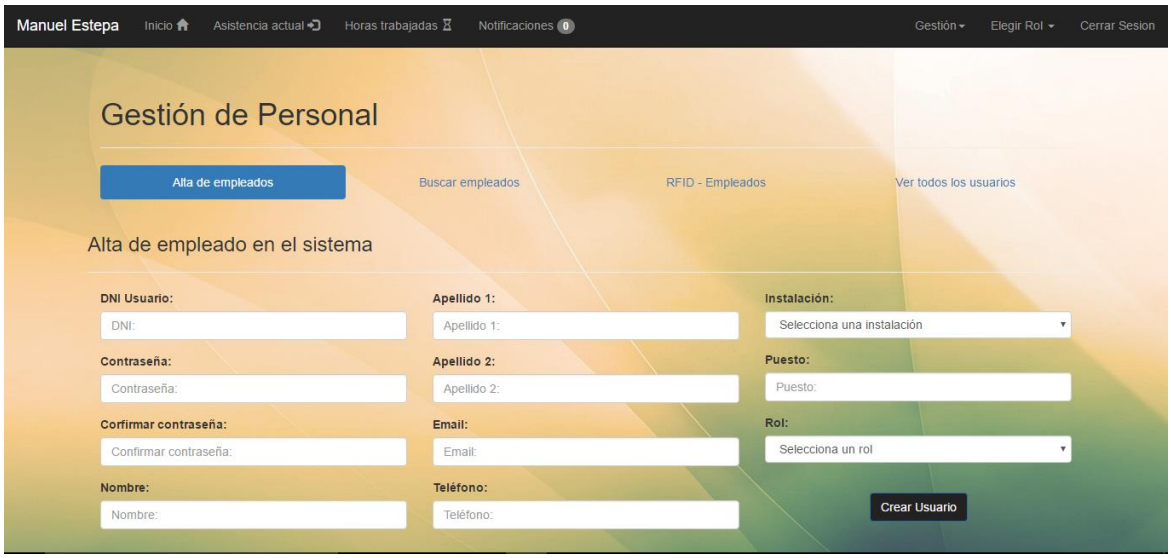


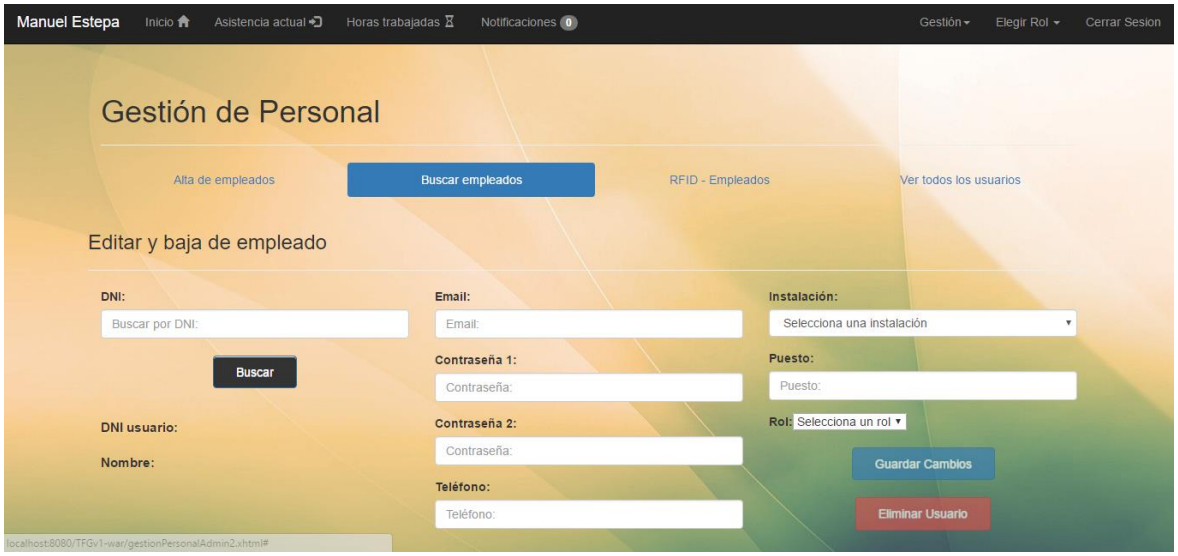


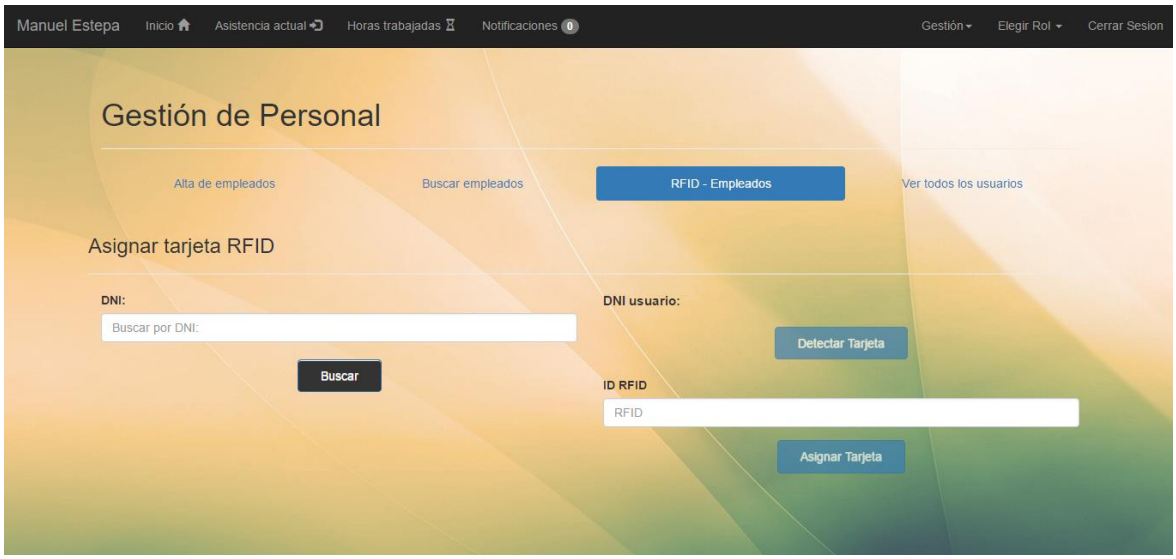


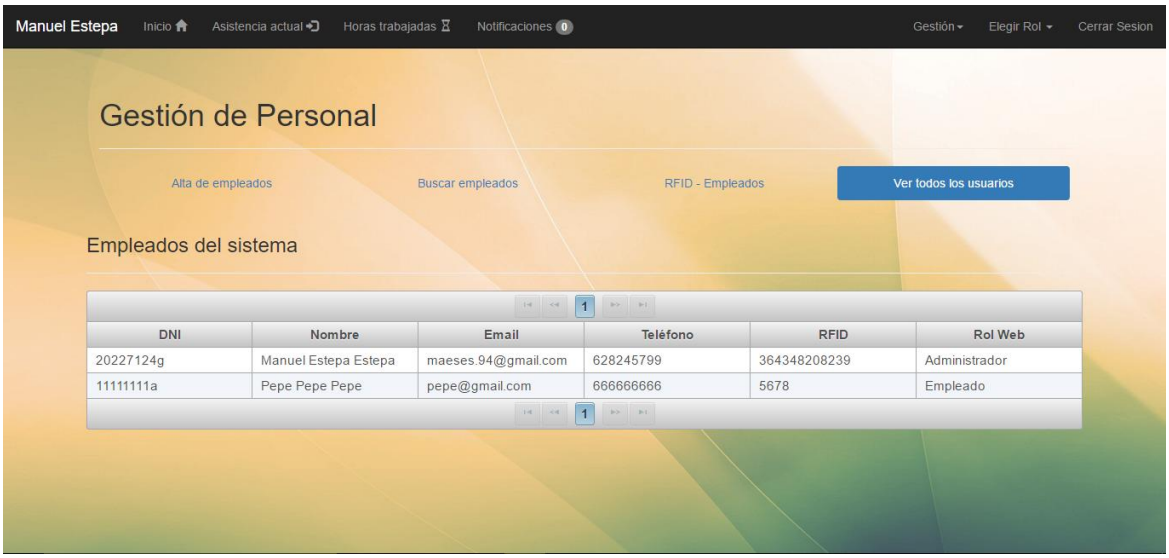

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 1
Gestión
Elegir Rol
Cerrar sesión
Notificaciones
Id
Empleado
Descripción
Fecha
1
Manuel Estepa (20227124g)
Prueba de notificaciones.
23/6/2016
Resolver


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 1
Gestión
Elegir Rol
Cerrar sesión
Resolver Notificaciones
Datos de la notificacion
Empleado:
20227124g
Fecha:
23/6/2016
Descripción:
Prueba de notificaciones.
Estado:
PENDIENTE
Resolver sin acción
Resolver notificación
Acción:
Seleccione una opción
Fecha (dd/mm/yyyy):
Hora (hh:mm:ss):
Instalación:
Resolver


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar Sesion
Gestión de Personal
Alta de empleados
Buscar empleados
RFID - Empleados
Ver todos los usuarios
Alta de empleado en el sistema
DNI Usuario:
DNI:
Contraseña:
Contraseña:
Confirmar contraseña:
Confirmar contraseña:
Nombre:
Nombre:
Apellido 1:
Apellido 1:
Apellido 2:
Apellido 2:
Email:
Email:
Teléfono:
Teléfono:
Instalación:
Selecciona una instalación
Puesto:
Puesto:
Rol:
Selecciona un rol
Crear Usuario


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar Sesion
Gestión de Personal
Alta de empleados
Buscar empleados
RFID - Empleados
Ver todos los usuarios
Editar y baja de empleado
DNI:
Buscar por DNI:
Buscar
DNI usuario:
Nombre:
Email:
Email:
Contraseña 1:
Contraseña:
Contraseña 2:
Contraseña:
Teléfono:
Teléfono:
Instalación:
Selecciona una instalación
Puesto:
Puesto:
Rol: Selecciona un rol
Guardar Cambios
Eliminar Usuario
localhost:8080/TFGv1-war/gestionPersonalAdmin2.xhtml#

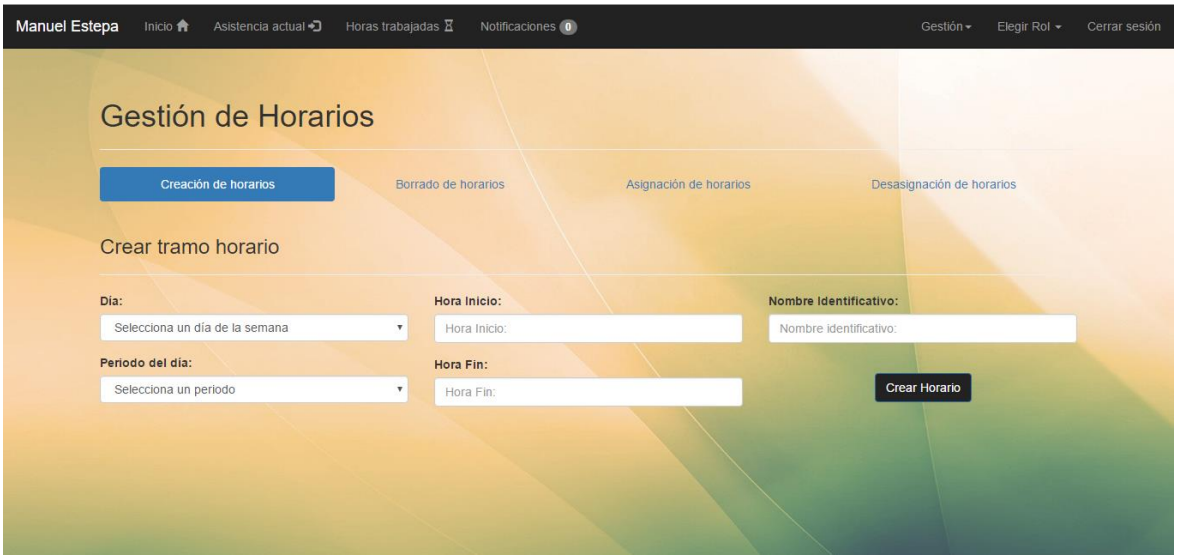

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar Sesion
Gestión de Personal
Alta de empleados
Buscar empleados
RFID - Empleados
Ver todos los usuarios
Asignar tarjeta RFID
DNI:
Buscar por DNI:
Buscar
DNI usuario:
Detectar Tarjeta
ID RFID
RFID
Asignar Tarjeta

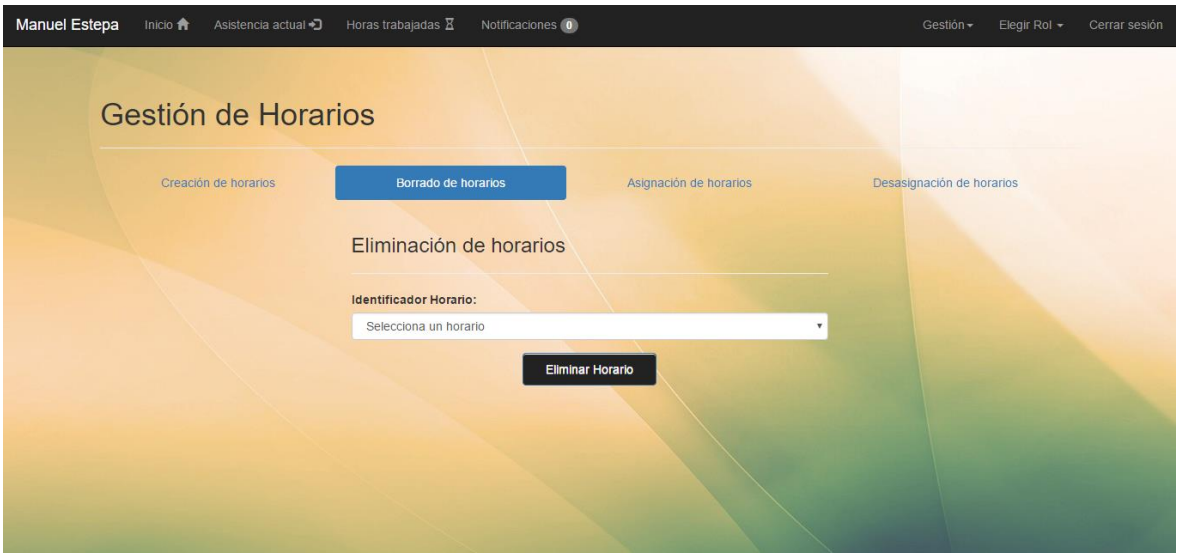

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar Sesion
Gestión de Personal
Alta de empleados
Buscar empleados
RFID - Empleados
Ver todos los usuarios
Empleados del sistema
1
DNI
Nombre
Email
Teléfono
RFID
Rol Web
20227124g
Manuel Estepa Estepa
maeses.94@gmail.com
628245799
364348208239
Administrador
11111111a
Pepe Pepe Pepe
pepe@gmail.com
666666666
5678
Empleado
1

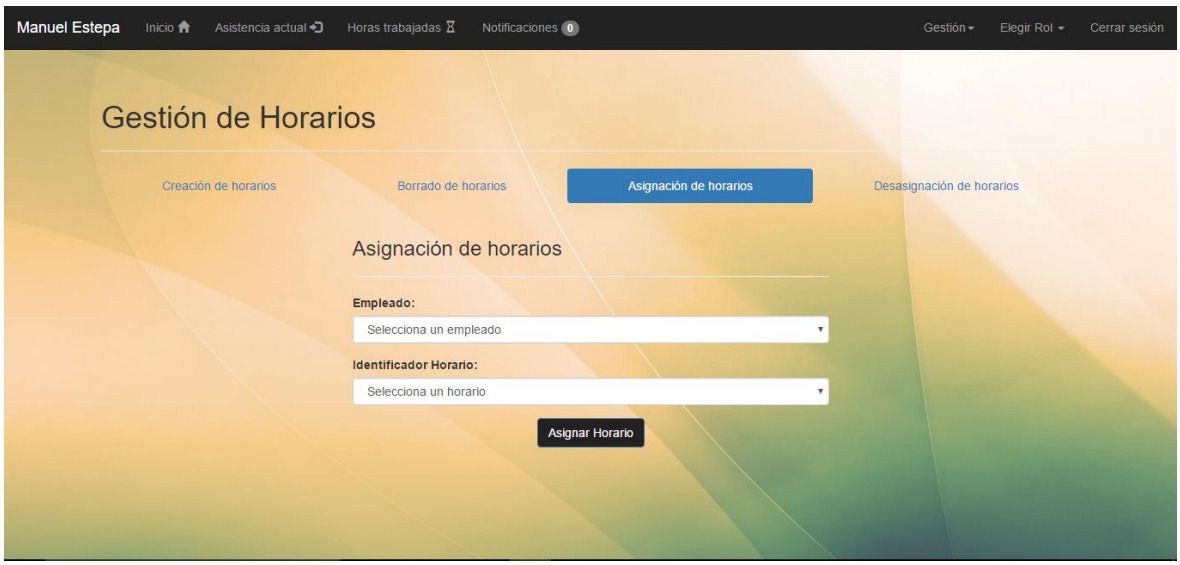

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Instalaciones
Creación de instalaciones
Edición y borrado de instalaciones
Ver todas las instalaciones
Crear instalación
Nombre:
Nombre
Ubicación:
Ubicación
Tipo de instalación:
Selecciona una opción
Crear Instalación

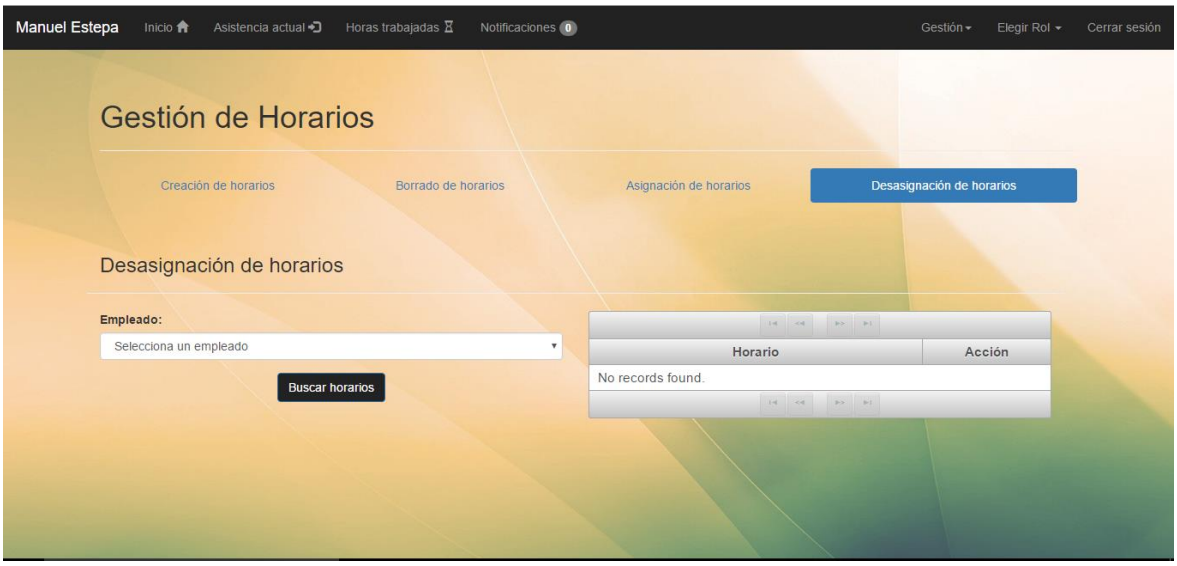

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Instalaciones
Creación de instalaciones
Edición y borrado de instalaciones
Ver todas las instalaciones
Edición y baja de instalaciones
Nombre:
Nombre:
Buscar
Guardar Cambios
Eliminar Instalacion
ID:
ID:
Nombre:
Nombre:
Tipo:
Selecciona una opción
Ubicación:
Ubicación:

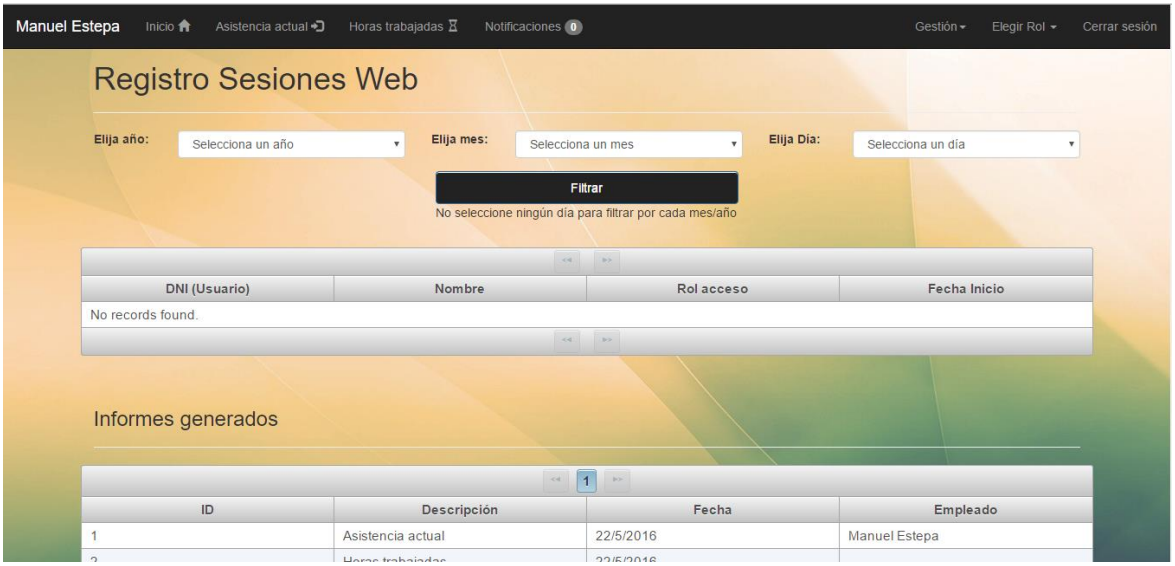

Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Instalaciones
Creación de instalaciones
Edición y borrado de instalaciones
Ver todas las instalaciones
Instalaciones del sistema
ID
Nombre
Tipo
Ubicación
1
Sede1
SEDE
Calle 1


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Horarios
Creación de horarios
Borrado de horarios
Asignación de horarios
Desasignación de horarios
Crear tramo horario
Día:
Selecciona un día de la semana
Hora Inicio:
Hora Inicio:
Nombre Identificativo:
Nombre identificativo:
Periodo del día:
Selecciona un periodo
Hora Fin:
Hora Fin:
Crear Horario


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Horarios
Creación de horarios
Borrado de horarios
Asignación de horarios
Desasignación de horarios
Eliminación de horarios
Identificador Horario:
Selecciona un horario
Eliminar Horario


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Horarios
Creación de horarios
Borrado de horarios
Asignación de horarios
Desasignación de horarios
Asignación de horarios
Empleado:
Selecciona un empleado
Identificador Horario:
Selecciona un horario
Asignar Horario


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Gestión de Horarios
Creación de horarios
Borrado de horarios
Asignación de horarios
Desasignación de horarios
Desasignación de horarios
Empleado:
Selecciona un empleado
Buscar horarios
Horario
Acción
No records found.


Manuel Estepa
Inicio
Asistencia actual
Horas trabajadas
Notificaciones 0
Gestión
Elegir Rol
Cerrar sesión
Registro Sesiones Web
Elija año:
Selecciona un año
Elija mes:
Selecciona un mes
Elija Día:
Selecciona un día
Filtrar
No seleccione ningún día para filtrar por cada mes/año
DNI (Usuario)
Nombre
Rol acceso
Fecha Inicio
No records found.
Informes generados
ID
Descripción
Fecha
Empleado
1
Asistencia actual
22/5/2016
Manuel Estepa