

SISTEMA TELEMÁTICO PARA EL REGISTRO, CONTROL DE INGRESO/SALIDA Y CONSULTAS DE TRAFICO DE MOTOCICLETAS EN LA FACULTAD TECNOLÓGICA DE LA UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

J.S. Martínez. G y W.D. Amaya. C

Resumen--En este artículo se presentará y explicará el prototipo para la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas, el cual busca facilitar la gestión referente al ingreso de motos al parqueadero y generar reportes de los datos correspondientes; mediante el uso de un sistema NFC; eso trabajando junto a un sistema de raspberry pi 2 unido a un servo motor que simulará el acceso del parqueadero, además el sistema se conecta a un servidor de forma remota con el fin de realizar el consumo de los diferentes servicios que se encuentran almacenados, todo esto realizado bajo los lineamientos de la metodología SCRUM agile, la cual permite obtener constantes entregables en cortos periodos de tiempo denominados Sprint.

Palabras clave-- BDD, MóduloXBee, NFC, Raspberry pi 2, RFID, Scrum Agile, Servo motor, Spring Boot MVC

Abstract-- This article will present and explain the prototype for the Technological Faculty of the Universidad Distrital Francisco José de Caldas, which seeks to facilitate the management of the entry of motorcycles into the parking lot and generate reports of the corresponding data; through the use of an NFC system; that working together with a system of raspberry pi 2 attached to a servo motor that will simulate the access of the parking lot, in addition the system is connected to a server remotely in order to realize the consumption of the different services that are stored, everything This is done under the guidelines of SCRUM agile methodology, which allows to

obtain constants deliverable in short periods of time called Sprint.

Keywords--BDD, XBee Module, NFC, Raspberry pi 2, RFID, Agile Scrum, Servo Motor, Spring Boot MVC

I. INTRODUCCIÓN

BAJO la constante observación que se realizó en el parqueadero de la facultad tecnológica de la Universidad Distrital Francisco José de Caldas se llegó a la conclusión que posee un gran inconveniente referente a la organización que es llevada, esto relacionado a la forma en la cual se realiza el ingreso de las diferentes motos, debido a que en las horas de mayor frecuencia de usuarios se forman largas filas para el ingresa y salida de los mismos.

¿Cuál sería una solución que perdure en el tiempo y realmente minimice de forma significativa las problemáticas mencionadas en cuanto al ingreso, salida y gestión de motocicletas, las cuales utilizan el parqueadero de la facultad tecnológica de la Universidad Distrital Francisco José de Caldas?

Por lo tanto se diseñó y se propuso un prototipo el cual mediante el uso de un sistema telemático de RFC permita agilizar este proceso, de igual forma generar reportes.

II. DESARROLLO DEL ARTÍCULO

Con el fin de dar una explicación clara referente al proyecto, es necesario realizar una descripción de los diferentes componentes utilizados dentro del mismo, comenzando por los componentes físicos del sistema telemático:

A. *NFC y RFID*

“La tecnología NFC (Near Field Communication) es uno de esos desarrollos aplicados al teléfono móvil que le permiten relacionarse, de modo interactivo, con otros dispositivos. En este caso, la interacción se realiza a través de un protocolo de comunicaciones compatible con la tecnología inalámbrica, basada en las Normas ISO 14443 A y B, ya en aplicación en el transporte público para la utilización de las tarjetas inteligentes sin contacto.

La implantación de las tarjetas inteligentes sin contacto ha precisado de importantes inversiones económicas en relación con los equipo de control de acceso, formas de pago y sistemas de gestión. NFC podría ayudar a optimizar estas inversiones ya realizadas.

NFC, en consecuencia, puede garantizar la compatibilidad con los distintos requisitos funcionales que las distintas aplicaciones de las tarjetas sin contacto ya implantadas exigen: seguridad anti vandálica y de datos personales, versatilidad, capacidad de crecimiento y compatibilidad con otras aplicaciones.

NFC es ya una realidad tecnológica y el transporte público una necesidad económica y social incuestionable. No es posible crecer económicamente sin un transporte público de calidad y la calidad tiene un coste nada despreciable.

Por otro lado, las necesidades de financiación del transporte público, imprescindibles para mejorar el servicio ofrecido, no pueden ser ajenas a las oportunidades que las llamadas nuevas tecnologías ofrecen.” [3]

B. *Raspberry pi2*

“Es una maravilla en miniatura, aguarda en su interior un importante poder de cómputo en un tamaño no más grande que el de una tarjeta de crédito.” [5]

C. *Python*

“Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollado rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python.”

“Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y en menor medida, programación funcional.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que quiere decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).”

D. *Spring Boot*

“Es un framework basado en java para la construcción de aplicaciones web y empresariales. A diferencia de otros marcos de trabajo (framework) spring boot proporciona una gran variedad de características que se ocupan de las necesidades empresariales modernas a través de sus proyectos de cartera.

Spring proporciona flexibilidad para configurar sus modelos de diferentes formas ya sea mediante XML, Anotaciones y JavaConfig. Con el número de características aumentadas, la complejidad también aumenta y la configuración se vuelve tediosa y propensa a errores, dado a eso el equipo de Spring creó Spring Boot para abordar la complejidad de la configuración.

Spring es muy popular debido a las siguientes razones: el enfoque de inyección de dependencias estimula la escritura de código fácil de probar, fácil de utilizar, junto a una potente capacidad de gestión de transacciones de bases de datos, simplifica en gran medida la integración con otros framework de java tales como JPA / Hibernate ORM, Struts / JSF y por último el uso del patrón de diseño de MVC.

Spring además posee una serie de módulos extras que le facilitan el cumplir con tareas mucho más robustas; de los

cuales resaltan spring data el cual facilita el acceso a datos desde almacenes de datos relacionales y NoSQL, spring security un robusto marco de seguridad para aplicaciones, spring integration una aplicación de Enterprise Integration Patterns para facilitar la integración con otras aplicaciones empresariales utilizando mensajería ligera y adaptadores declarativos, entre otros más.”[6]

E. MVC (Model View Controller)

“Es un patrón utilizada para aislar la lógica de negocio de la interfaz de usuario, el modelo representa la información de la aplicación y las reglas de negocio utilizadas para manipular los datos, la vista corresponde a elementos de la interfaz de usuario, como textos, campos de texto y demás, etc. Implicando la comunicación entre el modelo y la vista. El controlador se encarga de gestionar las acciones realizadas por el usuario.” [7]

F. Scrum Agile

Es importante mencionar el hecho de utilizar una metodología como Scrum agile para el desarrollo del proyecto y el uso de BDD (Behavior-DrivenDevelopment).

Para comenzar cabe aclarar que Scrum no es una metodología es un marco de trabajo, es decir, no te va a decir cómo hacer las cosas.

En gran medida Scrum se basa en el manifiesto ágil: “Estamos descubriendo formas mejores de desarrolla software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar: individuos e interacciones sobre procesos y herramientas software funcionando sobre documentación extensiva, colaboración con el cliente sobre negociación contractual y respuesta ante el cambio sobre seguir el plan. Esto es, aunque valoramos los elementos de la derecha valoramos más los de la izquierda.”[24]

Con el fin de profundizar más acerca de scrum es importante mencionar que posee con ciclos de trabajo los cuales son denominados sprint; usualmente los sprint tienen una duración de 1 a 2 semanas pero dependiendo de la complejidad del proyecto el periodo de tiempo puede variar hasta llegar incluso a 4 semanas en pocas ocasiones.

Ahora con el fin de tener una serie de tareas a realizar por parte del equipo de desarrollo existe un rol dentro del equipo que se encarga de gestionar y priorizar el orden de las diferentes tareas, esta categorización es realizada por el Product owner (PO) también conocido como dueño del producto este rol es el más cercano al cliente y conoce todas sus dolencias e inconvenientes que le genera el no tener el

producto, el objetivo del PO es recolectar cada uno de los requerimientos y necesidades del cliente y explicárselas al equipo de desarrollo / trabajo.

Seguidamente otro rol especial dentro de un equipo scrum es el rol del Scrum master (SM) el objetivo de este rol es quitar cualquier inconveniente que el equipo de desarrollo pueda tener referente a necesidad de algún material, recursos o permiso necesario; básicamente debe encargarse que el equipo de desarrollo realice únicamente su labor y no se distraiga por cosas externas al proyecto.

Posteriormente es necesario organizar los diferentes requerimientos de acuerdo a su prioridad de acuerdo a lo que genera algún tipo de ganancia para el negocio o proyecto, para ello el equipo scrum realiza una serie de reuniones que le permite tener siempre en claro cuáles son los objetivos a por los cuales se está apuntando, una de las más importantes es la reunión de Planning (reunión de planificación). Antes que nada antes del Planning el PO y SM se han reunido con el fin de organizar que tareas serán realizadas en el sprint; todas las tareas son anexadas como historias de usuario, en estas historias de usuario se realiza una descripción de la tarea con los diferentes criterios de aceptación (características que debe cumplirse a lo largo del sprint para dar por terminada la tarea), una vez que se tienen todas las historias de usuario se procede a realizar la planning junto al equipo de desarrollo; de acuerdo a la dificultad de las tareas el equipo de desarrollo (sin incluir al PO y al SM) miden las tareas asignándoles un “peso”, hay varias formas de asignar este peso, la más común es utilizando la serie de fibonacci (1,2,3,5,8...) es decir a una tarea que requiere de alrededor de tres (3) horas para ser realizada se le asigna un peso de 1, a una tarea que requiere alrededor de cinco(5) horas de trabajo se le asigna un peso de 2 y así sucesivamente. Es importante mencionar que esta no es una regla de oro la medición de los pesos para las tareas puede variar de acuerdo a la madurez del equipo Scrum y sus diferentes necesidades y formas de trabajo. Durante la reunión de planning el equipo de trabajo decide cuantas tareas va a realizar durante el sprint de acuerdo a los pesos asignados, el objetivo de asignarles un peso a las tareas permite calcular la “velocidad de trabajo” del equipo. Las historias de usuario que no alcanzan a entrar dentro del sprint quedan almacenadas dentro del product backlog para ser puestas en marcha en el siguiente sprint.

La siguiente reunión importante dentro de scrum es la review; consiste en una reunión que se da al final del sprint en la cual se encuentra el equipo de desarrollo, PO, SM, el cliente y los diferentes stakeholders, durante esta reunión el equipo de desarrollo muestra los diferentes avances que se realizaron a lo largo del sprint, la idea de la review es que el cliente pueda dar su punto de vista y opiniones referentes al

desarrollo y de esta forma dar una retro alimentación y aclarar cualquier tipo de duda que pueda haber por parte del equipo desarrollador.

Después de que se finaliza la review se realiza una reunión de retrospectiva, en esta reunión solo se encuentra el equipo scrum, el objetivo es realizar una retro alimentación interna del equipo la cual busca felicitar el trabajo realizado por parte del equipo y además identificar inconvenientes que se hayan presentado a lo largo del sprint con el fin de darles una posible solución o evitarlos completamente para los siguientes sprint.

Y por último para scrum agile es muy importante realizar un seguimiento constante a lo que hace cada uno de los miembros del equipo, para ello todos los días se hacen sesiones cortas denominadas daily en las cuales cada uno de los miembros del equipo cuanta las diferentes cosas que realizo el día anterior, lo que piensa y va a realizar el día de hoy y por ultimo comenta si tiene algún tipo de inconveniente que no le permita realizar sus labores.

G. BDD

Primero que todo mediante el uso de una metodología agile como scrum se logran obtener grandes avances en un proyecto, reduciendo en gran medida el tiempo que se debe ejercer con el fin de hacer cambios de funcionalidades o paradigmas debido a inconvenientes en el momento de definir el o los objetivos de in proyecto. De igual forma ayuda al equipo de desarrollo a construir y entregar un software con una calidad superior esto mediante el proceso de seleccionar correctamente las funcionalidades o especificaciones que realmente generan un valor para el negocio en este caso para el desarrollo de una aplicación y además que cada uno de los miembros que componen el equipo de trabajo conozca los diferentes requerimientos del sistema que se desea construir; permitiendo de esta forma que no se desperdicie el tiempo del equipo de trabajo en cosas que no generan valor o que posteriormente solo se convertirán en algún tipo de estorbo.

“BDD no es una metodología de desarrollo de software por derecho propio. No es un reemplazo para Scrum, XP, Kanban, RUP, o cualquier metodología que esté usando actualmente. BDD incorpora, construye y mejora ideas de muchos de estas metodologías. Y no importa qué metodología esté usando, hay maneras en que BDD puede ayudar a hacer su vida más fácil.”[20]

BDD también tiene en consideración las razones por las cuales suelen fracasar los proyectos de software; las razones más comunes suelen ser:

- Porque el software no se construyó de la manera correcta
- Porque el software que se construyó no es el correcto

BDD es de gran ayuda en el momento de realizar el análisis de requerimientos debido a que se busca que los requerimientos sean desglosados de todo tipo de lenguaje técnico hasta llegar al punto de que sean fácilmente comprendidos por cualquier individuo (cabe mencionar que BDD también permite crear escenarios para desarrollar pruebas de aceptación automatizadas). Con el fin de llegar a este desglose de los diferentes requerimientos se crean “escenarios” que se basan en la estructura de Gherkin.

Gherkin “es un lenguaje específico para el negocio que permite describir el comportamiento del software sin detallar en cómo se implementa ese comportamiento, posee dos propósitos: documentación y pruebas automatizadas.

Al igual que Python y YAML, Gherkin es un lenguaje orientado a líneas que utiliza la sangría para definir la estructura. Terminaciones de línea terminan las sentencias (por ejemplo pasos). Se pueden utilizar espacios o tabs para realizar los espaciados. La mayoría de las líneas comienzan por una palabra clave.” [4]

La estructura de un archivo Gherkin normalmente luce de la siguiente manera:

```
Funcionalidad: Un texto corto pero descriptivo de lo que se desea

Descripción funcionalidad: Descripción textual del valor de negocio de esta cara característica
    Las reglas de negocio que gobiernan el enfoque de la funcionalidad
Y algún tipo de información adicional para facilitar el entendimiento de la funcionalidad

Escenario: Determinada situación

Dado alguna precondition
Y alguna otra precondition
Cuando alguna acción realizada por el actor
Y alguna otra acción
Entonces resultado esperado
```

Un ejemplo real seria:

```
Feature: User who wants to complete the form
    To get a credit card
```

As a interested person in a credit card
 I want to be able enter in a form and do my request

Scenario: request a credit card as a pre approved customer

Given that Brayan wants get a credit card
 When he complete the initial form with cedula 1234567 and date expedition 1-Enero-2017
 And he accept the legal terms
 And he answer the questions of authentication
 And he fill his email addresstest@mail.com
 And he check his data in the captured form
 Then the window show a message of Felicitaciones

H. MATERIALES

A continuación se muestran los componentes físicos utilizados para el desarrollo del sistema telemático, módulo NFC.

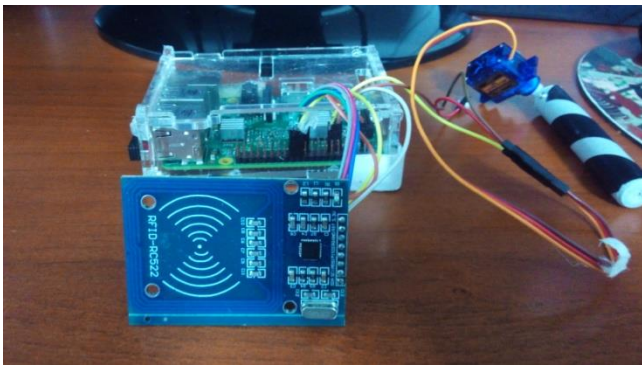


Ilustración 1 Módulo NFC

Seguidamente la Raspberry pi 2



Ilustración 2 Raspberry pi 2

Seguidamente del servo motor

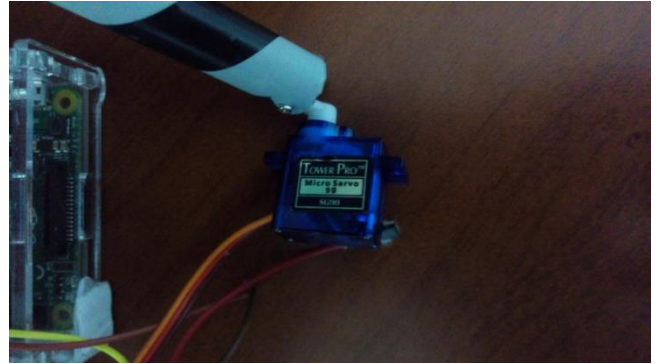


Ilustración 3 Servo Motor

Así es como luce la integración de todos los componentes.



Ilustración 4 Integración componentes físicos

I. RESULTADOS

A continuación se presentan los desarrollos implementados con el fin de dar solución a lo planteado al inicio del proyecto, primero que todo se presenta el diagrama de la base de datos.

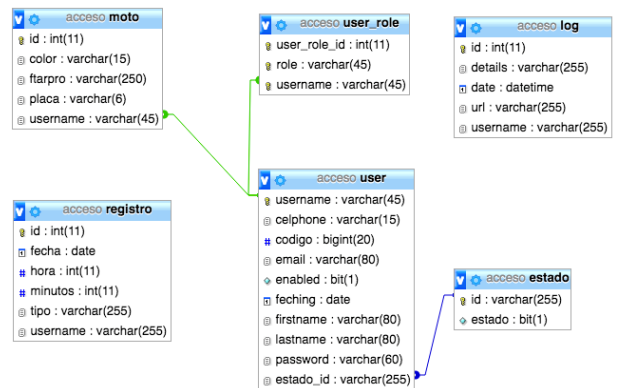


Ilustración 5 Diagrama de la base de datos

Seguidamente de la plataforma que se desarrolló con el fin de consumir los diferentes servicios, cabe

mencionar que la plataforma también sirve para dispositivos móviles sin ningún tipo de conflicto.

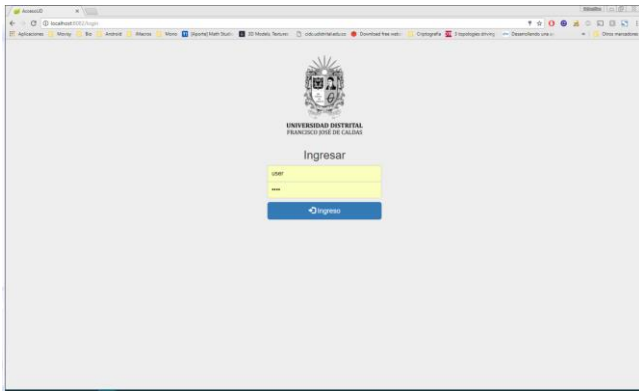


Ilustración 6 Login de la aplicación

Resumen general de los usuarios registrados para utilizar el servicio del parqueadero de la facultad junto a la visualización en tiempo real de las motos que se encuentran en el parqueadero.

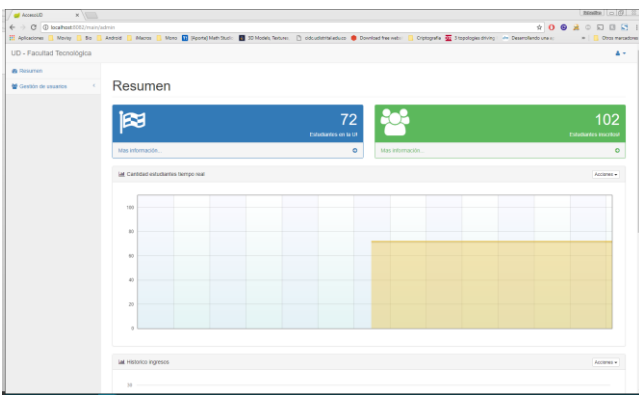


Ilustración 7 Resumen de la aplicación parte 1

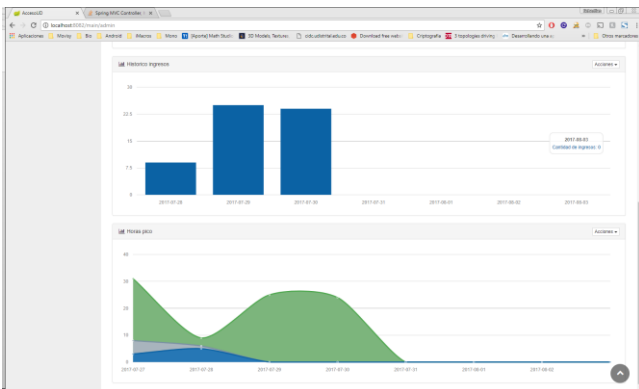


Ilustración 8 Resumen de la aplicación parte 2

Para cada una de las gráficas se puede generar los reportes pertenecientes a las motos como un archivo de imagen .jpg

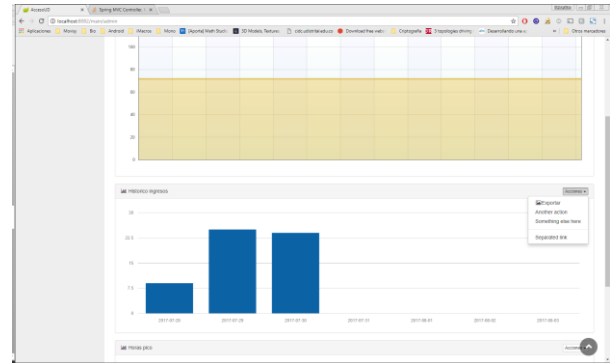


Ilustración 9 Exportar graficas

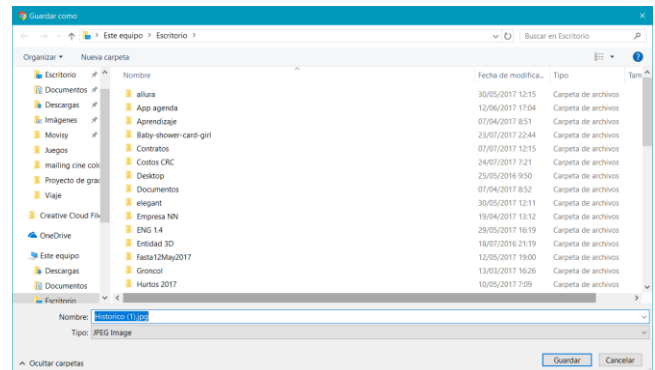


Ilustración 10 Guardar graficas / histórico de datos

Posteriormente en el menú lateral del lado izquierdo dentro de la opción de Gestión de estudiantes se podrán encontrar a cada uno de los usuarios registrados a través de la aplicación

Acciones	Moto	Datos
[Icon]	[Icon]	Código: 38102876001 Nombre: EDUARDO OSORIO EMAIL: OSORIOE@univalcaldas.edu.co
[Icon]	[Icon]	Código: 38102876002 Nombre: EDITH BASTIEN EMAIL: BASTIENEDITH@gmail.com
[Icon]	[Icon]	Código: 38102876003 Nombre: EDUARDO OSORIO EMAIL: OSORIOE@univalcaldas.edu.co
[Icon]	[Icon]	Código: 38102876004 Nombre: EDUARDO OSORIO EMAIL: OSORIOE@univalcaldas.edu.co
[Icon]	[Icon]	Código: 38102876005 Nombre: EFRAN LOPEZ EMAIL: LOPEZEFRA@gmail.com
[Icon]	[Icon]	Código: 38102876006 Nombre: EFRAN LOPEZ EMAIL: LOPEZEFRA@gmail.com
[Icon]	[Icon]	Código: 38102876007 Nombre: EDUARDO OSORIO

Ilustración 11 Usuarios registrado

En donde se pondrá visualizar la información ingresada para cada uno de los usuarios

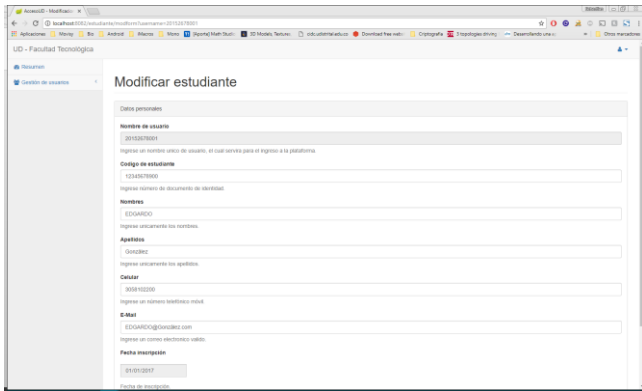


Ilustración 12 Módulo de modificación - datos del estudiante

Y lo más importante el estado de activo o inactivo para el uso del parqueadero

Estado



Ilustración 13 Campo de texto estado del usuario

Dentro de la gestión de los estudiantes se le podrá agregar las motos a cada uno de los usuarios, en donde se selecciona el color de la moto, placa y se agrega un archivo digital perteneciente a la tarjeta de propiedad

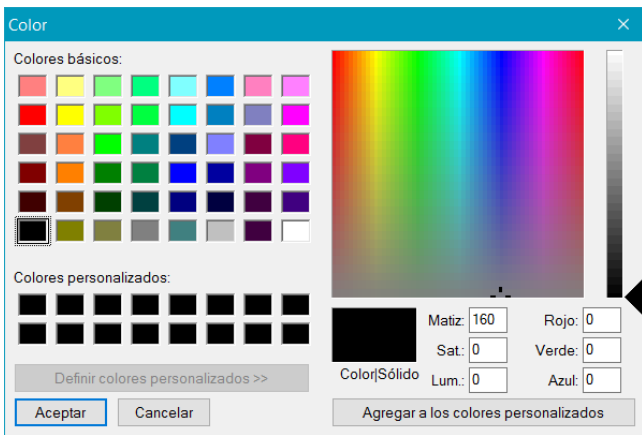


Ilustración 14 Input especial para seleccionar el color de la moto

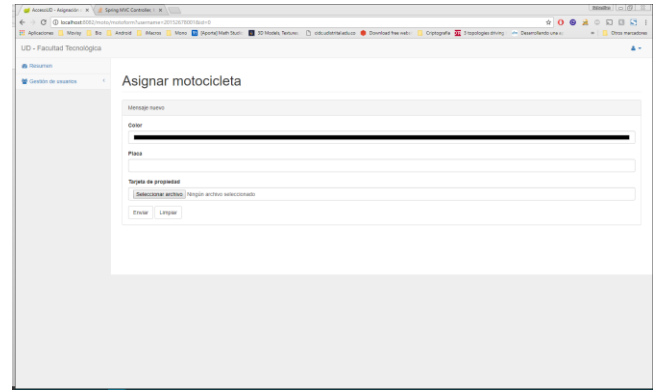


Ilustración 15 Formulario de asignación de moto a un usuario ya existente

Posteriormente se podrá visualizar que la operación fue realizada con éxito

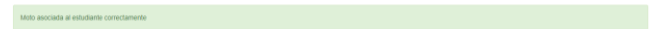


Ilustración 16 Confirmación exitosa de asociación de moto

En la parte inferior de la sección de gestión de estudiantes se encuentra un botón que nos permitirá ingresar a los nuevos usuarios

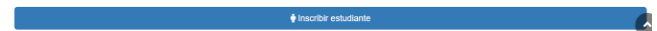


Ilustración 17 Botón de creación de nuevo estudiante

Mediante el cual se accederá al formulario de registro

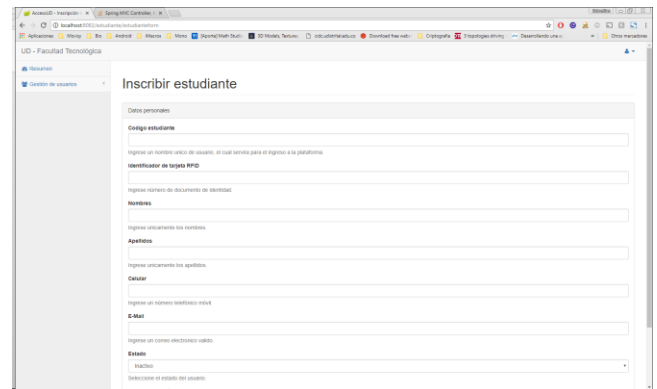


Ilustración 18 Formulario de registro de estudiantes

III. CONCLUSIONES

- Se logran desarrollar mediante Python y utilizando un sistema que permite la lectura de las tarjetas inteligentes.
- Se plantea y se construye el prototipo en el cual se demuestra el funcionamiento de la aplicación y puesta en marcha del mismo.

- Se aplican y se profundizan los conocimientos adquiridos a lo largo de la carrera y de igual forma se abarcan nuevos sectores de conocimiento mediante la implementación y uso de otro tipo de plataformas con las cuales no se había tenido contacto a lo largo de lo transcurrido en la facultad.
- Se cumple a cabalidad con las diferentes historias de usuario y con las prácticas dadas por la metodología de Scrum Agile.
- Finalmente se logra dar por completado cada uno de los objetivos que se plantearon al inicio del proyecto.

REFERENCIAS

- [1] XBee cl, «¿Qué es XBEE?», 2017. [En línea]. Available: <http://xbee.cl/que-es-xbee/>. [Último acceso: 23 Agosto 2017].
- [2] © Asoc. "Foro de Nuevas Tecnologías en el Transporte, ITS España", LIBRO BLANCO SOBRE LA APLICACIÓN DE LA TECNOLOGÍA NFC EN EL TRANSPORTE PÚBLICO, Madrid: C/ Henri Dunant, 2013.
- [3] J. F. Ruiz., «LA TECNOLOGIA NFC SE ACERCA A LA BIBLIOTECA,» RUIDERAE: Revista de Unidades de Información, vol. 2, n° 6, p. 7, 2014.
- [4] A. Hansson, «Gherkin,» 19 Enero 2017. [En línea]. Available: <https://github.com/cucumber/cucumber/wiki/Gherkin>. [Último acceso: 15 Julio 2017].
- [5] G. H. Eben upto, «Raspberry Pi - Guía del usuario,» Raspberry Pi.
- [6] S. P. R. Katamreddy, «Why Spring Boot?,» 20 Mayo 2016. [En línea]. Available: <https://dzone.com/articles/why-springboot>. [Último acceso: 17 Agosto 2017].
- [7] rj45, «Simple Example of MVC (Model View Controller) Design Pattern for Abstraction,» 8 Abril 2008. [En línea]. Available: <https://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>. [Último acceso: 11 Agosto 2017].
- [8] Python Software Foundation., «Tutorial de Python - Introducción,» 2017. [En línea]. Available: <http://docs.python.org.ar/tutorial/3/real-index.html>. [Último acceso: 12 Julio 2017].
- [9] toptechboy, «RASPBerry PI LESSON 28: CONTROLLING A SERVO ON RASPBERRY PI WITH PYTHON,» 2 Junio 2015. [En línea]. Available: <http://www.toptechboy.com/raspberry-pi/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/>. [Último acceso: 25 07 2017].
- [10] Wikipedia, «Raspberry Pi,» 20 Julio 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Raspberry_Pi. [Último acceso: 23 Julio 2017].
- [11] Hetpro Store, «Módulo RFID-RC522 RF con Arduino UNO SPI,» 25 Abril 2014. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>. [Último acceso: 23 Julio 2017].
- [12] A. Castro, «¿Qué es Raspberry Pi, dónde comprarla y cómo usarla?,» 23 Enero 2014. [En línea]. Available: <http://computerhoy.com/noticias/hardware/que-es-raspberry-pi-donde-comprarla-como-usarla-8614>. [Último acceso: 23 Julio 2017].
- [13] C. Á. Caules, «¿Qué es Spring Boot?,» 8 Marzo 2016. [En línea]. Available: <http://www.arquitecturajava.com/que-es-spring-boot/>. [Último acceso: 23 Julio 2017].
- [14] Wikipedia, «Maven,» 28 Junio 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/Maven>. [Último acceso: 23 Julio 2017].
- [15] C. Á. Caules, «Spring MVC Configuración,» 20 Noviembre 2013. [En línea]. Available: <http://www.arquitecturajava.com/spring-mvc-configuracion/>. [Último acceso: 23 Julio 2017].
- [16] Wikipedia, «Python,» 22 Julio 2017. [En línea]. Available: <https://es.wikipedia.org/wiki/Python>. [Último acceso: 23 Julio 2017].
- [17] M. d. C. García, «Programar posiciones en un Micro Servo Tower Pro SG90 9G,» 13 Enero 2016. [En línea]. Available: <http://miarduinounotieneunblog.blogspot.com.co/2016/01/programar-posiciones-en-un-micro-servo.html>. [Último acceso: 23 Julio 2017].
- [18] Wikipedia, «Thymeleaf,» 24 Marzo 2015. [En línea]. Available: <https://es.wikipedia.org/wiki/Thymeleaf>. [Último acceso: 23 Julio 2017].
- [19] M. L. Michelone, «Raspbian: Sistema operativo gratuito para la Raspberry Pi,» 2 Agosto 2012. [En línea]. Available: <https://www.unocero.com/2012/08/02/raspbian-sistema-operativo-gratuito-para-la-raspberry-pi/>. [Último acceso: 23 Julio 2017].
- [20] J. Garzas, «Enlace permanente a Entendiendo qué es BDD (Behavior-Driven Development) (I),» 15 Diciembre 2014. [En línea]. Available: <http://www.javiergarzas.com/2014/12/bdd-behavior-driven-development-1.html>. [Último acceso: 30 Julio 2017].
- [21] Toptechboy, «RASPBerry PI LESSON 28: CONTROLLING A SERVO ON RASPBERRY PI WITH PYTHON,» 2 Junio 2015. [En línea]. Available: <http://www.toptechboy.com/raspberry-pi/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/>. [Último acceso: 26 Julio 2017].
- [22] martijnschut, «RC522 with Raspberry Pi 2,» 4 Abril 2016. [En línea]. Available: <http://www.martijnschut.nl/rc522-with-raspberry-pi-2/>. [Último acceso: 26 Julio 2017].
- [23] Informatica Moderna, «TARJETA DE RED INALÁMBRICAS,» 2017. [En línea]. Available: http://www.informaticamoderna.com/Tarjetas_red_inalam.htm.
- [24] M. B. A. v. B. A. C. W. C. M. F. J. G. J. H. A. H. R. J. J. K. B. M. R. C. M. S. M. K. S. J. S. D. T. Kent Beck, «Manifiesto por el Desarrollo Ágil de Software,» 2001. [En línea]. Available: <http://agilemanifesto.org/iso/es/manifiesto.html>. [Último acceso: 10 Septiembre 2017].

Wilber David Amaya Cruz nació en Bogotá, Colombia, el 10 de Junio de 1993. Se graduó como bachiller técnico en



Artes gráficas del Instituto Técnico Industrial Centro don Bosco en el año 2009 posteriormente como tecnólogo de sistematización de datos en la Universidad Distrital Francisco José de Caldas el 15 de Mayo del 2015. Ejerce actualmente en la

fábrica digital del Banco Colpatría del grupo Scotiabank Cra. 7 #24-51, Bogotá.

Joan Sebastian Martinez Galvis nació en Bogotá, Colombia, el 19 de Agosto de 1992. Se graduó como



bachiller técnico en Gestión de sistemas de manejo ambiental del I.E.D. Gustavo Restrepo en el año 2009 posteriormente como tecnólogo de sistematización de datos en la

Universidad Distrital Francisco José de Caldas el 15 de Mayo del 2015. Ejerce actualmente como independiente, con emprendimiento de empresa de soluciones tecnológicas en la ciudad de Bogotá.