Carwash business remote monitoring platform using magnetic sensors.

José De Jesús Garfias López
garfiaslopez@gmail.com

Arturo Daniel Jauregui López
adjl9509@gmail.com

Jafeth Ascensión Alonso Carreón
jaalonso@ipn.mx

Gabriela Sánchez Meléndes
gsanchezgsm@yahoo.com

Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, Instituto Politécnico Nacional
Ingeniería en Comunicaciones y Electrónica. Ciudad de México.

***Resumen*—Se describe el proceso de desarrollo de una plataforma web y móvil para la administración de negocios de auto lavado enfocado en la conectividad vía remota, su principal función es el conteo de vehículos usando sensores de campo magnético, al ingresar un auto al negocio se envía una señal al punto de venta (que es una computadora con el software) y el operador de la caja procede a cobrar el vehículo, toda la información esta disponible en la nube para que el dueño del negocio pueda consultarla mediante una aplicación móvil y web, con esto se logra reducir el margen de pérdida en las ganancias por la incorrecta administración de los operadores de la caja en el negocio y dar información útil al dueño del negocio acerca del estado financiero del negocio en tiempo real, con el objetivo que pueda tomar mejores decisiones sobre el mismo.**

I. Introducción

La administración es fundamental para cualquier negocio ya que de esta depende el incrementar las ganancias a un margen controlado de pérdidas, con el avance tecnológico se tiene a disposición muchos programas de computadora que ayudan con esta labor tan esencial. Es importante que el programa de computadora entienda a la perfección las necesidades del negocio para que se pueda generar los mejores resultados y así tener un mejor aprovechamiento de recursos y mayores ganancias, debido a esto, es necesario enfocar el software de computadora hacia una vertical de negocio especifica, teniendo siempre en mente las reglas de negocio especificas al programar, logrando no caer en lo general. Logrando tener módulos muy específicos que ayudan al desarrollo de la interfaz de usuario para lograr que sea lo más simple e intuitiva posible.

II. Descripción de la problemática: Auto lavados

Los auto lavados son negocios en los cuales los clientes pagan por un servicio físico sobre el vehículo, el cual consiste en lavado, pulido y encerado, es por norma que en la mayoría de los negocios de auto lavados se tiene que esperar una cantidad moderada de tiempo (20 minutos aproximadamente) y se paga alrededor de $70 pesos mexicanos por esta labor, los negocios de este tipo son muy importantes para los clientes que no desean perder tiempo haciendo esta tarea ellos mismos, y prefieren pagar por un servicio más eficiente y rápido.

Datos de la Procuraduría Federal del Consumidor (PROFECO), revelan que, al segundo semestre de 2015, existían 23,851 unidades económicas en México para el servicio de lavado y encerado de automóviles y camiones. El Estado de México, Veracruz y Puebla son los que concentran el mayor número, de los cuales la mayoría no tiene un sistema de administración por computadora.

TABLA I
Numero de auto lavados en México

Estado	Unidades de auto lavado
Estado de México	2,231
Veracruz	2,018
Puebla	1,667
Jalisco	1,457
Michoacán	1,205
Ciudad de México	1,147

Fuente: Elaboración propia con datos de la PROFECO. [1]

Los auto lavados deben de cumplir con al menos 4 etapas fundamentales de servicio para cada auto, el enjabonado, el enjuague, el secado y por ultimo el aspirado, con estas 4 etapas podemos afirmar que se ha dado un servicio completo al vehículo.

Existen dos tipos de auto lavados los de tipo cajón y tipo túnel, el primer tipo son los más comunes y funcionan distribuyendo los automóviles que entran al terreno del negocio en cajones parecidos a los de estacionamiento, donde el cliente deja su vehículo para que se pueda proceder con las etapas, existen al menos 2 empleados por cajón que se encuentran realizando el servicio y la capacidad máxima del negocio de autos que puede atender al mismo tiempo es directamente proporcional al numero de cajones que tenga disponibles el terreno, la media de cajones son de alrededor unos 7 cajones.

El tipo túnel basa su funcionamiento en un túnel donde el cliente ingresa su auto pero no desciende de su vehículo, sino que espera a que la primera etapa del lavado se realice, una vez que finaliza la primera etapa el cliente avanza su vehículo por el túnel, la segunda etapa se realiza, lo avanza una tercera vez y espera que la tercera etapa concluya, estas primeras tres etapas se pueden hacer con personal humano o con la ayuda de máquinas industriales especializadas para esta tarea, en la ultima etapa el cliente estaciona su auto en algún cajón y se procede con la ultima etapa. La capacidad de autos que se pueden lavar en paralelo va a depender de la cantidad de cajones disponibles para la ultima etapa, más 3 de los espacios disponibles en el túnel.

III. Administración del negocio

El tiempo que se tarda en llevar el proceso completo de lavado en el negocio debe de ser el mínimo, por esta razón los empleados tienen que trabajar bajo presión dependiendo la demanda de vehículos a determinada hora, y como resultado solo hay una persona que se encuentra en caja realizando el cobro a los clientes por el servicio, estos operadores de caja deben de estar al pendiente de los carros que entran y que tipo de modelo son para poder seleccionar la tarifa correcta para el tipo de auto y cobrar lo justo, en ocasiones existe el error humano y no se anotan los vehículos que se cobran o se cobran a diferente tarifa de la original, estos errores se reflejan al final del día, en el corte de caja cuando el operador de caja debe de entregar las ganancias del día al dueño y asegurarse que las ganancias se encuentren en balance, muchas veces sobra o falta dinero que nadie sabe en donde se encuentra, propiciando así una fuga de efectivo por la mala administración de los operadores de caja.

IV. Solución al problema

El sistema de administración de auto lavados que se propone soluciona los problemas ocasionados anteriormente mediante la introducción de sensores y un punto de venta para ayudar al operador de la caja a realizar un cobro mas exacto.

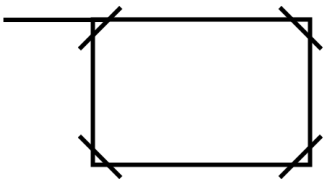


Ilustración 1 Diagrama de posición de la bobina.

Se instala en el piso de cada cajón que existe en el auto lavado un sensor de lazo magnético (o detección de masa vehicular), para poder detectar masas metálicas cuando estas se encuentran en el cajón.

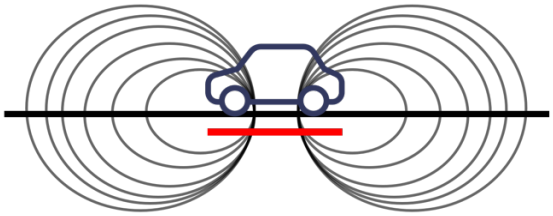


Ilustración 2 Representación de los campos magnéticos de la bobina.

Si un vehículo pasa por arriba del sensor, un pulso es enviado al punto de venta que detona la apertura de un ticket que indica al operador de caja que tiene que cobrar ese vehículo, inmediatamente después de la apertura el operador tiene que seleccionar que tipo de vehículo es para poder asignar la tarifa que el operador debe de cobrar al cliente del vehículo, con esto el operador solo tiene que cobrar la cantidad que le indica el sistema y no tiene que preocuparse por estar al pendiente de los autos que entran al negocio, esto ayuda a que al final del día las cuentas de la caja sean más acertadas y se evite fugas de ganancias.

Además, el sistema esta conectado a un servidor y una base de datos donde se almacenan todos los registros de compras y operaciones para que sean consultados mediante una aplicación web, y el dueño no tenga la necesidad de ir físicamente al negocio para saber como van sus ventas del día o cuanto dinero hay disponible en caja.

Adicionalmente se agregan otras funcionalidades de administración tales como, el manejo de inventario del almacén, conceptos de gastos e ingresos, administración de sucursales y usuarios.

V. Arquitectura de la plataforma

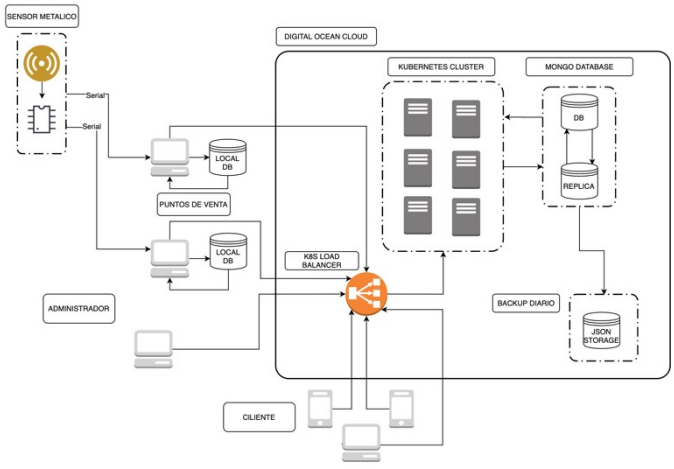


Ilustración 3 Diagrama de Arquitectura del Sistema.

La arquitectura del software se divide en:

- **Hardware**: Sensor de masa vehicular.
- **Software**: Aplicación de escritorio de punto de venta.

- **Back-end y bases de datos**: manejo de peticiones y guardado de información en la nube.
- **Front-end**: Administrador web responsivo donde se consultan los datos de los negocios.

Cada etapa se desarrolla con diferentes tecnologías que interactúan entre si para lograr el conjunto de procesamiento y almacenaje de datos.

VI. Hardware

El hardware esta diseñado para ser controlado mediante una tarjeta de desarrollo Arduino[2], la cual se programa para detectar cuando se activa el detector de masa vehicular y mediante comunicación serial enviar esta señal al punto de venta.

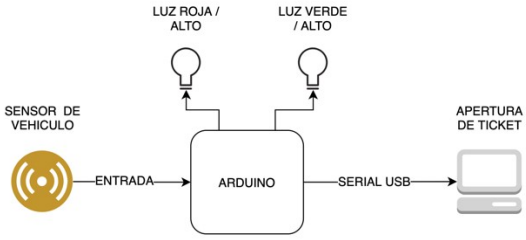


Ilustración 4Diagrama de circuito de control para la detección de masas vehiculares.

El circuito de masa vehicular que se utiliza es comercial y se puede encontrar como **Wejoin WJLD110**.

Ilustración 5Circuito detector de masa vehicular.

A el se conecta la bobina inductiva que se instala en el piso de donde se desea instalar el sensor de vehículos, adicionalmente tiene dos relevadores, uno normalmente abierto y otro normalmente cerrado que se activan cuando la bobina es excitada por una placa metálica, estos relevadores ayudan a cerrar un flujo de corriente que funciona como pulso para el Arduino que en su programación cuando detecta este pulso envía por comunicación serial un carácter al punto de venta que interpreta como la apertura de un nuevo ticket a cobrar.

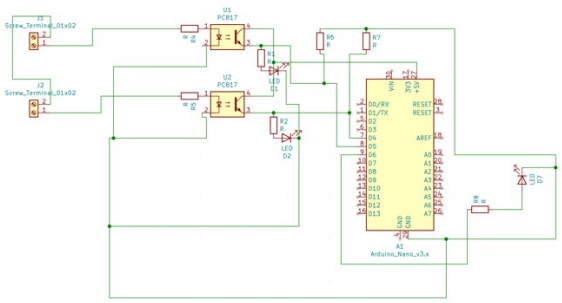


Ilustración 6Diagrama electrónico del circuito de control.

El circuito también controla el encendido de relevadores, a los cuales van conectados los semáforos que van instalados en el túnel del auto lavado que funcionan como indicadores hacia el cliente de lo que tiene que hacer, cuando el semáforo esta en verde es que puede avanzar y en rojo es que necesita detenerse para iniciar el proceso de lavado, el semáforo siempre se encuentra en verde hasta que se detecta un pulso proveniente del circuito de masa vehicular, lo que indica que el vehículo ya se encuentra en la zona deseada y se necesita detener el vehículo por lo tanto el relevador cambia de estado para encender el otro color de semáforo.

VII. Software: Punto de venta

El punto de venta es una aplicación de escritorio multiplataforma desarrollada con base en Electron JS [3], el cual es una librería que permite hacer aplicaciones nativas para varios sistemas operativos con una misma base de código, y esta base de código se desarrolla con tecnologías web, html, css y javascript, adicionalmente el framework de javascript que se utiliza es ReactJS [4]. Con estas tecnologías se desarrolla una sola aplicación que se exporta a sistema operativos como Linux, Windows y Mac. Lo que ayuda mucho al tiempo de desarrollo porque se evita el desarrollar código particular para cada sistema, se evita la curva de aprendizaje del lenguaje que utiliza cada sistema operativo. También ayuda a que el mantenimiento del código sea mucho mas fácil y el añadir nuevas funcionalidades sea mucho mas rápido, adicionalmente ya que son tecnologías web que muchas personas dominan, encontrar personas que se integren al proyecto o puedan aportar código se vuelve una tarea mas fácil.

Una funcionalidad muy importante es la capacidad del software de seguir funcionando sin la necesidad de conexión a internet, ya que es común que en los negocios el internet pueda tener fallas, lo que el programa tiene que hacer es guardar todas las transacciones realizadas a la base de datos que se intentaron hacer de forma local, y en cuanto el sistema detecte que existe internet, enviar toda esa información para sincronizar la información con la de la nube.

Para lograr eso se tiene que instalar una base de datos local, en este caso otra instancia de la base de datos MongoDB [5] pero con menos recursos que corre de forma local en el puerto

por default de la computadora, teniendo la base de datos local podemos guardar la información primero en esa base de datos, posteriormente tratar de enviar esa misma información al servidor mediante una petición HTTP gracias a la REST API que esta ejecutándose en el servidor, si la petición fue exitosa y se guardó la información en la nube, todo esta bien, en la base de datos local marcamos que ese registro fue guardado como exitoso, y todo continua normalmente. Lo interesante sucede cuando la petición no es exitosa (puede ser que se desconectó el internet o el servidor tiene problemas) y se tiene que actualizar en la base de datos local ese registro que no se guardó en el servidor de la nube, inmediatamente se detona un subproceso que revisa cada 10 segundos si existe conexión a internet, con la finalidad de que cuando se recupera la conexión a internet se envíen todos los datos de la base de datos local marcados como fallidos de guardado en el servidor de la nube. Una vez que en algún intento logró enviar los datos, este subproceso es eliminado y todo sigue funcionando normalmente.

VIII. BACK-END Y BASES DE DATOS

El backend es una aplicación desarrollada en Node JS que funciona como REST API, la cual tiene funcionalidades de creación, edición, actualización y borrado para cada entidad de la base de datos, mediante urls y métodos HTTP.

Node JS es una librería que ayuda a construir aplicaciones del lado del servidor con Javascript, tiene la ventaja que es asíncrono y muy eficaz al momento de procesar múltiples llamadas de servidor al mismo tiempo sin bloquear ninguna ejecución, lo que permite mayor velocidad de respuesta en las peticiones de información.

Como motor de base de datos se utiliza MongoDB la cual es una base no relacional orientada a guardar grandes cantidades de información en el rango de los millones dependiendo las características del servidor sobre el cual se ejecute, y esta información se almacena como objetos del tipo JSON, es comúnmente usada para BigData gracias a su velocidad de consulta sobre grandes cantidades de información, se aprovecha mejor cuando la información a almacenar no tiene muchas relaciones entre si y no se necesitan consultas muy complejas entre tablas.

Adicionalmente, MongoDB tiene la posibilidad de ejecutarse como una base de datos distribuida a la cual se le pueden configurar replicas que guardan una copia de la información en tiempo real para tener alta disponibilidad y evitar perdidas de información, como un respaldo a largo plazo cuenta también con una utilería MongoDump para hacer copias completas de la información en toda la base de datos cada cierto tiempo y guardarla como un solo archivo BSON que se puede respaldar en algún repositorio en la nube para después recuperar.

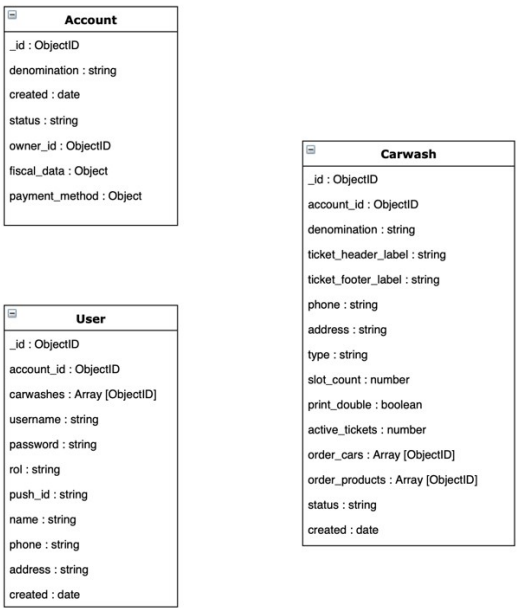


Ilustración 7 Tablas de entidades principales de la base de datos.

Como entidades principales tenemos "Account", "User", "Carwash" para modelar el sistema, donde se tiene como entidad de mayor prioridad "Account" a donde van a pertenecer "User" y "Carwash", el concepto de cuenta es para separar las diferentes franquicias de auto lavados que quisieran darse de alta en el sistema, posteriormente "Users" y "Carwash" donde "Carwash" representa una sucursal de la franquicia y los usuarios pueden pertenecer a mas de una sucursal pero no a mas de una cuenta, los usuarios separados por cuenta y tienen varios niveles, "Admin", "Operator", "Employee", donde cada uno tiene privilegios diferentes al momento de editar las otras entidades de la base de datos.

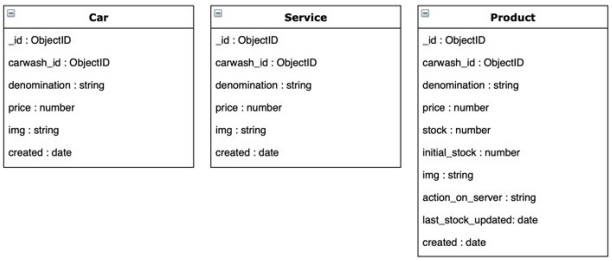


Ilustración 8 Tablas de entidades secundarias que dependen directamente de la sucursal.

En las entidades "Car", "Service" y "Product" son las entidades secundarias que dependen directamente de "Carwash" y son los servicios y productos que se ofrecen en la misma, la relación esta de esa forma por que estos servicios y productos pueden ser diferentes entre sucursales de auto lavados y/o pueden tener diferentes precios, entonces se tiene separado por sucursales.

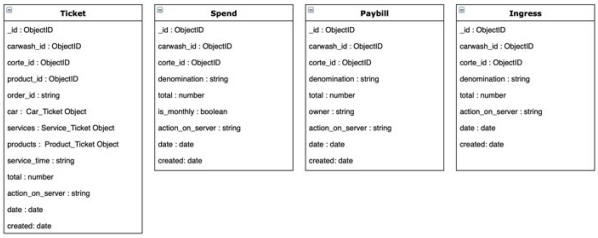


Ilustración 9 Tablas de entidades generadas por movimientos del día a día en los negocios.

Las entidades de “Ticket”, “Spend”, “Paybill” e “Ingress” son entidades que se generan por los movimientos que se hacen en el punto de venta diariamente y que ayudan a la contabilidad del negocio generando información que pertenece a un “Corte” que es la entidad que ayuda a agrupar esa información, ya que por requerimientos del negocio esas entidades no se agrupan siempre por día sino por un periodo de tiempo entre que se abre y se cierra el negocio y estos no dependen directamente del día.

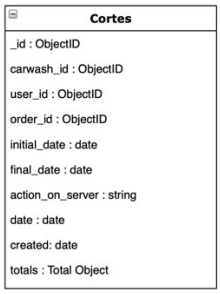


Ilustración 10 Tabla de la entidad corte de la base de datos.

Puede existir mas de un corte realizado en el día que agrupa las entidades anteriores y sirven para agruparse y sacar los totales y porcentajes de ventas que se han hecho en el negocio, ya sea por día o por un intervalo de tiempo.

IX. Front-end

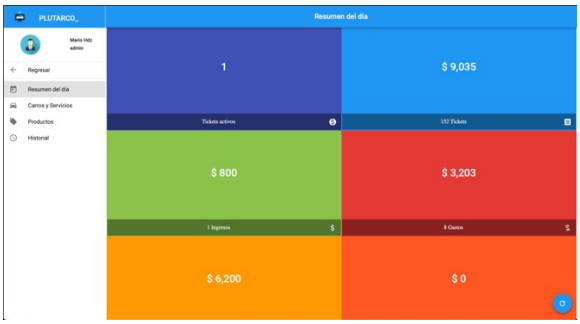


Ilustración 11 Dashboard de administración pantalla principal

El administrador que funciona como manejador de contenidos es una interfaz web que esta realizada sobre ReactJS y que consulta y modifica la información mediante la REST API del backend, con el objetivo de poder dar de alta entidades como sucursales, personal, asignar servicios, precios, productos a las sucursales, etc. Así como también mostrar un dashboard que muestra los datos generados por los negocios en cualquier día o intervalo de tiempo, a modo que se puedan tomar decisiones de negocio.

El modelo orientado a componentes de ReactJS ayuda mucho al tiempo de desarrollo ya que permite la reutilización del código sin muchas complicaciones, ya que al crearse un componente que funciona, se puede implementar en diferentes vistas, a su vez como la aplicación de software de punto de venta esta realizado con ReactJS, se pueden compartir componentes entre plataformas con la misma base de código sin complicaciones, lo que incrementa aun mas el tiempo de desarrollo.

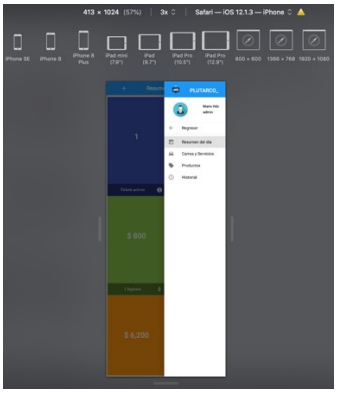


Ilustración 12 Interfaz adaptativa para dispositivos móviles

Como librería de estilos se utiliza Material Design [6] el cual es una guía desarrollada por Google en la cual indica como deberían ir diseñados lo componentes y como debe de ser su interfaz con el usuario, es el mismo diseño que ellos utilizan para todas sus plataformas y que años de estudio han dado fruto, para poderlo adaptar y usar con ReactJS se utiliza la librería React Material UI [7], que brinda una gama de componentes ya prestilizados para uso libre.

X. Algunas Equivocaciones Comunes

Debido a que el sensor de masa vehicular detecta cualquier objeto que sea metálico, puede ser activado mediante alguna placa metálica, o un recogedor de metal, una pala, esto puede afectar al conteo de los vehículos que se lleva en el negocio, para poder corregir esto se puede ajustar la sensibilidad del sensor para que la masa de metal se muy grande para poder ser detectada.

Debido a que la opción de seleccionar el tipo de vehículo queda abierta en el sistema para que el operador de caja pueda elegir, esta expuesta a errores como por ejemplo que el operador seleccione un tipo de vehículo por otro que no es y así cobrar una cantidad errónea, por lo que a futuro se planea implementar un sistema de visión artificial que pueda detectar los tipos de vehículos de forma automática y así evitar esta posible falla en el sistema.

XI. Referencias

[1] "Lavados de autos. Limpieza sobre ruedas", *gob.mx*, 2016.[En línea]. Disponible: https://www.gob.mx/profeco/documentos/lavados-de-autos-limpieza-sobre-ruedas. [Consulta: 01- Feb- 2020].

[2] "The Making of Arduino", *IEEE Spectrum: Technology, Engineering, and Science News*, 2011. [En línea]. Disponible: https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino. [Consulta: 01- Feb- 2020].

[3] "Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS.", *Electronjs.org*, 2013. [En línea]. Disponible: https://www.electronjs.org. [Consulta: 01- Feb- 2020].

[4] "React – Una biblioteca de JavaScript para construir interfaces de usuario", *ReactJS*, 2013. [En Línea]. Disponible: https://es.reactjs.org. [Consulta: 01- Feb- 2020].

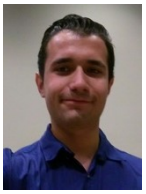
[5] "The most popular database for modern apps", *MongoDB*, 2009. [En Línea]. Disponible: https://www.mongodb.com.[Consulta: 01- Feb- 2020].

[6] "Google Material Design", *Material Design*, 2014. [En Línea]. Disponible: https://material.io/design/. [Consulta: 01- Feb- 2020].

[7] "React Material UI", *Material UI*, 2014. [En Línea]. Disponible: https://material-ui.com/es/. [Consulta: 01- Feb- 2020].

XII. Biografías

José de Jesús Garfias López: Es pasante de Ingeniería en Comunicaciones y Electrónica, con especialidad en computación de la Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Zacatenco del Instituto Politécnico Nacional. Áreas de interés: desarrollo web, desarrollo móvil multiplataforma, arquitectura de sistemas.



Arturo Danial Jauregui López: Es pasante de Ingeniería en Comunicaciones y Electrónica, con especialidad en computación de la Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Zacatenco del Instituto Politécnico Nacional. Áreas de interés: sistemas embebidos, internet de las cosas, desarrollo web.

M. en C. Jafeth A. Alonso Carreón.- Es Maestro en Ciencias en Ingeniería de Telecomunicaciones e Ingeniero en Comunicaciones y Electrónica, ambos por la ESIME del Instituto Politécnico Nacional. Actualmente es profesor titular en la Academia de Computación del Departamento de Ingeniería en Comunicaciones y Electrónica en la ESIME unidad Zacatenco del IPN. Áreas de interés: Procesamiento Digital de Señales, Electromagnetismo Computacional, internet de las cosas, visión por computadora.

M. en C. Gabriela Sánchez Meléndez. Maestra en Ciencias en Ingeniería de Telecomunicaciones de la Sección de Estudios de Posgrado e Investigación (SEPI) del Instituto Politécnico Nacional (IPN). Actualmente es Profesora titular de la Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Profesional Adolfo López Mateos (ESIME). Áreas de Interés: Fibras Ópticas, Telecomunicaciones, Aplicaciones de Software, Visión por Computadora.