

DISEÑO DE UN SISTEMA BIOMÉTRICO PARA CONTROL DE ASISTENCIAS EN AMBIENTES UNIVERSITARIOS

DESIGN OF A BIOMETRIC SYSTEM FOR ATTENDANCE CONTROL IN UNIVERSITY ENVIRONMENT

Aldonso Becerra Sánchez

Universidad Autónoma de Zacatecas, México
a7donso@uaz.edu.mx

Emmanuel de Jesús Velásquez Martínez

Universidad Autónoma de Zacatecas, México
zefeca27@gmail.com

Gustavo Zepeda Valles

Universidad Autónoma de Zacatecas, México
gzepeda_@hotmail.com

Uriel Ramírez García Correa

Universidad Autónoma de Zacatecas, México
uramirezgc@uaz.edu.mx

Santiago Esparza Guerrero

Universidad Autónoma de Zacatecas, México
chago@uaz.edu.mx

Recepción: 22/octubre/2019

Aceptación: 23/noviembre/2019

Resumen

Actualmente las tareas de un profesor universitario requieren una relación entre la sesión de clase presencial y el registro de asistencia del estudiante, por lo que un mecanismo convencional para registrar dicha actividad de asistencia es capitalizado en una hoja de papel. Esta actividad deja de lado al alumno, además del tiempo que se consume. Este trabajo tiene como objetivo el diseño de un sistema asequible basado en biometría dactilar para el registro de asistencia del estudiante por cada curso al que asiste. La infraestructura desarrollada para este tipo de sistema requiere portabilidad y usabilidad para el usuario. El producto del control será obtenido por medio de un reporte de asistencias de huella digitalizada dactilar con mecanismo fiable de seguridad. Se dispone adicionalmente de una aplicación de escritorio desarrollada en Python que brinda funciones de gestión de usuarios.

Palabras Claves: Huella dactilar digital, lector de huella digital, Raspberry PI, registro de asistencia biométrica.

Abstract

Nowadays the tasks of a university professor require a relationship between the face-to-face class session and the student's attendance record, so that a conventional mechanism to record such attendance activity is capitalized on a sheet of paper. This activity leaves aside the student, in addition to the time consumed. The aim of this work is to design an affordable system based on fingerprint biometrics for the registration of student attendance for each course attended. The infrastructure developed for this type of system requires portability and usability for the user. The product of the control will be obtained by means of a digital fingerprint assistance report with a reliable security mechanism. It also has a desktop application developed in Python that provides user management functions.

Keywords: *Digital fingerprint, fingerprint reader, Raspberry PI, biometric attendance record.*

1. Introducción

Es habitual que en la mayoría de las instituciones educativas la asistencia de un alumno a sus sesiones de clase en sus diferentes asignaturas es una condición importante para tener derecho a evaluaciones parciales y finales. El requisito de asistencia regular significa que el alumno debe estar presente durante al menos el 80% del tiempo de clase. También significa que el estudiante debe estar presente en la institución durante todo el período durante el cual se está impartiendo el curso. Los maestros tienen el derecho de excluir a los estudiantes de los exámenes para esa materia si no están en clases el 80% del tiempo.

El trabajar en otro tema externo nunca es una justificación aceptable para la ausencia de clase. Sólo se aceptan razones médicas y otras circunstancias fuera de control directo del interesado. Además de asistir a sesiones presenciales, es obligatorio venir preparado para la clase y contribuir activamente a los ejercicios de resolución de problemas, discusiones y análisis de casos.

En algunos casos, la participación activa en la clase es un elemento de la evaluación.

Por tanto, el sistema de registro de asistencia evita que se presenten varias anomalías respecto a cada caso señalado en las que puede incurrir la perspectiva del curso del estudiante. El presente trabajo trata de brindar un sistema que permita, de manera activa y con movilidad, el registro de asistencias presenciales de estudiantes por medio de la interacción directa de él con el módulo desarrollado. Sistemas gestores de contenidos de aprendizaje como Moodle [Moodle, 2019] tiene subsistemas que permiten realizar procesos de validación de asistencia (adicionalmente existen otros como Blackboard), sin embargo, estos sistemas se basan en el hecho de que el control lo realiza directamente el maestro a través de la plataforma. Sin embargo, este sigue siendo un proceso manual.

Diversos trabajos relacionados a este tema se han ido desarrollando para ir solventando tareas similares y limitaciones de estos procesos manuales o semi-automáticos que implican registro de asistencias en entornos académicos. Por ejemplo, en el Instituto Tecnológico de Soledad Atlántico se diseñó e implementó un prototipo para el control de acceso a aulas y laboratorios, así como un registro automático de asistencia a clases. Este proceso pretendía reducir el tiempo que estas tareas conllevan. El sistema completo consiste en el desarrollo de software que soporta toda la gestión de recursos y asistencia, los equipos RFID (Radio Frequency IDentification) y demás hardware. Esta infraestructura controla el acceso a los espacios físicos y la comunicación entre los puntos de acceso en cada salón y el controlador central (Arduino) mediante ZigBee [Márquez, 2014].

En la Universidad Nacional Montemayor de San Marcos se propuso brindar una solución informática para el control de la asistencia de los alumnos. Este servicio se ofreció mediante una suscripción (Software as an Service), haciendo uso de la tecnología de lectura de la huella dactilar para la autenticación y registro de asistencia. Así mismo se brindó el uso de los Servicios Web para la comunicación y transferencia de información entre los componentes del sistema [Jiménez, 2018]. Algo similar se desarrolló en la Universidad De Córdoba, con una plataforma web para el control de asistencia de estudiantes, docentes y administrativos de la

Universidad en la sede Lórica. Este proceso se realizó mediante el uso de tarjetas inteligentes, dispositivos móviles y SMS [Sepúlveda, 2014].

En un ámbito más generalizable, en la Universidad de Extremadura (Centro Universitario de Mérida) se diseñó y desarrolló un sistema informático que permite (a través de sensores de huella dactilar) registrar la entrada y salida de profesores a sus clases y a las tutorías oficiales validadas por el departamento. Los fichajes se llevan a cabo a través de una aplicación web simple y corporativa. Esta información queda registrada en un servidor de base de datos, permitiendo generar diversos reportes de asistencias y faltas [Calle, 2016].

Por otro lado, en la Universidad Veracruzana se propuso el registro de asistencia de los estudiantes mediante reconocimiento facial (usando Matlab), apoyando al docente con la automatización de esta labor administrativa. Este trabajo ofreció una alternativa para las instituciones educativas con un elevado número de alumnos de diferentes niveles. Esto facilita el registro de asistencia a clases de los estudiantes de una manera autónoma y sistematizada [Benavides, 2016].

El sistema propuesto en este trabajo tiene como intención brindar un mecanismo controlado y no susceptible a falsificación de identidad o suplantación de procesos de asistencia mediante un sistema biométrico de huella digital. La idea es plasmar una tecnología institucional, móvil, económica y que brinde solución a situaciones de problemáticas manuales o semiautomáticas de control de asistencias.

2. Métodos

Para el desarrollo de este sistema se optó por una metodología ágil conocida como *Prototyping*. Esta metodología se define como un método de desarrollo de sistemas en el que un prototipo es construido, probado y luego reinventado según sea necesario. El proceso continúa hasta que finalmente se logre un prototipo aceptable a partir del cual el sistema o producto completo puede ser ahora desarrollado [Rouse, 2005]. Este modelo de prototipo permite obtener *feedback* de algunos de los interesados en el sistema en etapas tempranas. Esto permite que funcionalidades del sistema sean descartadas o suprimidas mientras que nuevas funcionalidades y necesidades fueran agregadas conforme van siendo requeridas.

El prototipo es una versión preliminar, intencionalmente incompleta o reducida de un sistema [Weitzenfeld, 2007]. El uso de prototipos es una herramienta útil para aplicarse en casi todas las actividades de diseño y creación de software.

Para el caso de las cuestiones tácticas del presente trabajo, el prototipo diseñado se basará en las especificaciones de los requerimientos bajo los supuestos de:

- Obtención de requerimientos del usuario.
- Validación de los requerimientos a través de la funcionalidad del producto.

El enfoque *Prototyping* sigue un proceso que se repite en cada iteración [Sommerville, 2000], [Becerra, 2018] (figura 1). Durante el establecimiento de los objetivos del prototipo, el fin es el seleccionar una de las funcionalidades a desarrollar y definir hasta donde llegaría el prototipo en etapa de evaluación. Esta tarea requiere de la selección cuidadosa de características que requieran ser evaluadas por el usuario final para su aprobación o desecho. Cada uno de estos pasos nos permite mejorar el sistema, incluso antes de que esté finalizado.

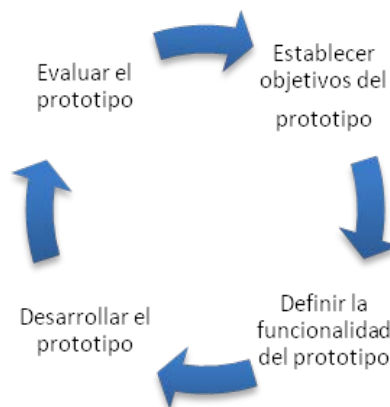


Figura 1 Ciclo de vida del prototipo.

Establecer Objetivos del Prototipo

El sistema de registro de asistencias por medio de lector biométrico se basa en las necesidades presentadas por parte de la Universidad Autónoma de Zacatecas. Por lo que fue de vital importancia realizar un listado de requerimientos que permitieran cumplir con las expectativas solicitadas. Para ello, inicialmente se dispuso a llevar a cabo los siguientes requerimientos:

- Sistema único con la finalidad de evitar problemas futuros al incorporarlo como parte de otros sistemas que actualmente se están desarrollando.
- Sistema con portabilidad para el traslado del módulo entre área de trabajo.
- Los módulos del sistema deben ser adaptables a los posibles cambios que se proyectan dentro del sistema de gestor de contenidos académicos.
- La aplicación total debe ser amigable para los usuarios finales.
- Se debe poder dar seguimiento a las asistencias y faltas del alumnado, generando un reporte día a día de las sesiones de las clases registradas.

Definir la Funcionalidad del Prototipo

El sistema proyectado deberá brindar la siguiente funcionalidad (figura 2):

- Disponer de un servidor de base de datos Oracle. Se debe alojar la información de un gestor de contenidos de aprendizaje: datos de alumnos, profesores, horarios, ciclos escolares, materias, trabajos y calificaciones. Específicamente para el módulo abordado se debe hacer énfasis en los datos escolares del alumno y la relación con el historial de asistencias.

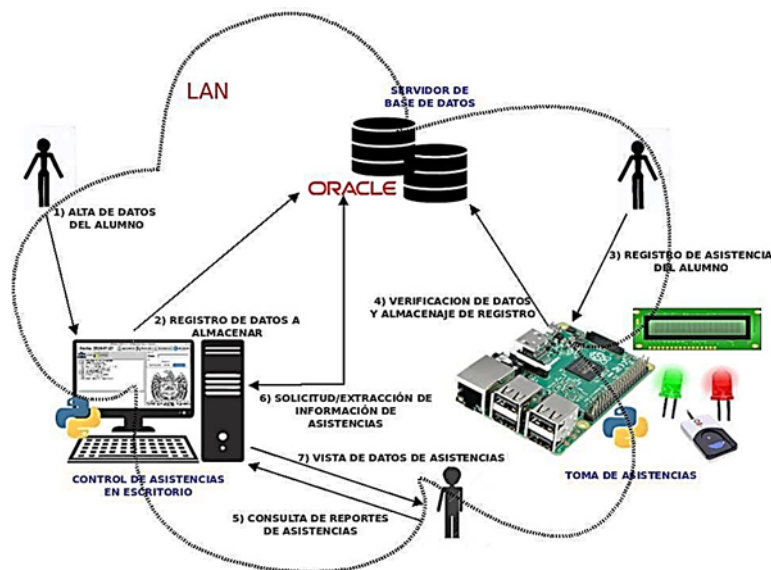


Figura 2 Flujo de datos del esquema de prototipo desarrollado.

- Disponer de un módulo de control del alumnado y reportes de asistencia. La aplicación debe ofrecer un panel de control para que los docentes tengan la

facilidad de gestionar las asistencias del alumno. Donde la funcionalidad principal es proporcionar cuatro tipos de operaciones crear, leer, actualizar y eliminar datos [CRUD, 2019]. Además, debe permitir realizar un seguimiento de los alumnos inscritos en cada clase, generando de esta forma reportes grupales o individuales de los alumnos y sus asistencias.

- Disponer de un módulo de registro de asistencia. Este módulo debe estar diseñado y desarrollado para funcionar bajo la Raspberry Pi y un lector de huella dactilar. Esta minicomputadora de bajo costo y de tamaño compacto permite ser ideal para interactuar con el alumno. La importancia de usar este tipo de arquitectura permite flexibilizar la movilidad y portabilidad del módulo desarrollado, al mismo tiempo que se brinda escalabilidad.

Desarrollar el Prototipo

Para el desarrollo del prototipo se especifican los elementos principales de cada uno de los módulos requeridos en la funcionalidad definida en el proyecto. En la figura 2 se observan los tres componentes principales del sistema: Servidor de base de datos, módulo de control de asistencias vía escritorio, y el módulo portátil de registro de asistencias. Cabe señalar que estos componentes se comunican entre ellos a través de la red local cableada Ethernet.

El flujo de datos inicia con el alta de los datos personales generales del alumno (paso 1 y 2); así como su huella dactilar, que se usará como base para la comparación en procesos subsecuentes. En esta misma etapa se dan de alta las asignaturas y grupos en los que el alumno quedará registrado (incluyendo horarios). En el paso 3 se dispone del módulo que permite realizar el registro de asistencia de manera interactiva por parte del alumno ante el subsistema portátil de la Raspberry. Este proceso continúa con la validación (paso 4) de que el alumno se encuentre dado de alta en un grupo en ese horario y que cumpla con los lineamientos de asistencia o retardo. Si no existe error, el proceso cumple la cometido de almacenar que se cumple con la estancia correspondiente en la sesión de clase, tanto al inicio como al fin de dicha situación. En el paso 5 se permite que el profesor pueda realizar consultas sobre las asistencias individuales o grupales de los cursos que imparte.

En el pasó 6 y 7 se extrae la información y se proporciona la vista de reportes.

Servidor de base de datos

El servidor de base de datos montado es Oracle Database 11g Enterprise Edition [Oracle Database, 2019]. Es una base de datos que contiene las respectivas tablas relacionadas para el sistema final. Este gestor es utilizado para la funcionalidad de otros módulos de un sistema global de contenidos de aprendizaje. Sin embargo, para fines de este trabajo, sólo se estará utilizando la estructura de las tablas necesarias para el alojamiento de datos de alumnos, materias, profesores, ciclos escolares, horarios, grupos y asistencias (figura 3). Se puede apreciar que una asignatura propensa a recibir asistencias de alumnos pertenece a un ciclo escolar. Estas asignaturas son cargadas por un determinado alumno y a un grupo específico abierto de esa materia. Dicho grupo tiene un horario definido y que es impartido por cierto profesor.

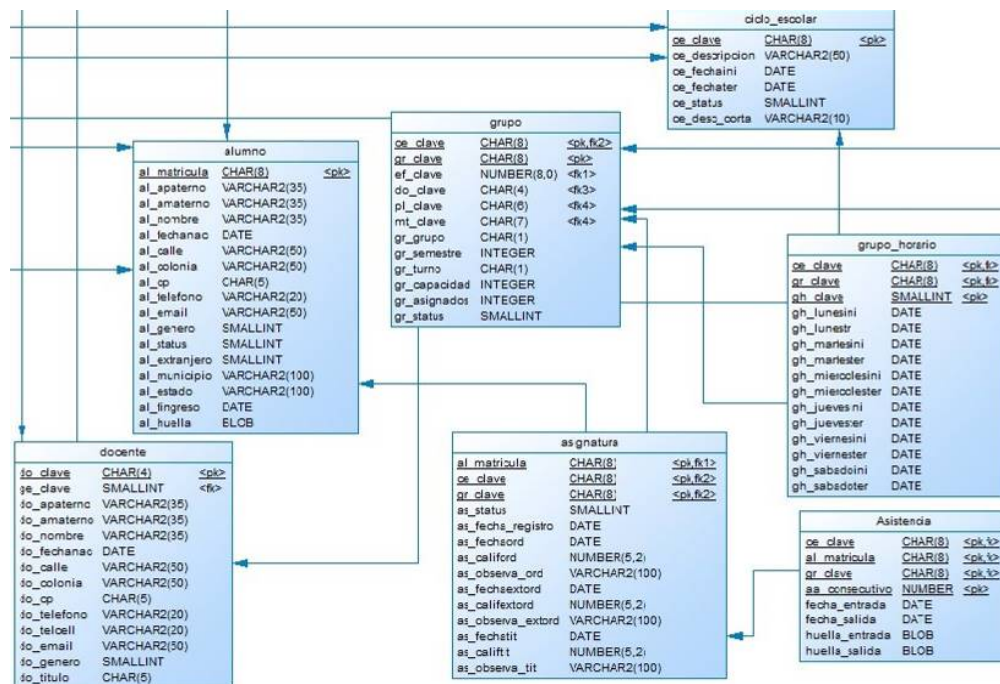


Figura 3 Esquema relacional de la sección de la base de datos usada en el prototipo.

El módulo desarrollado utiliza JDBC, la cual es una interfaz de programa de aplicación (API) empaquetada con la edición de Java SE. JDBC hace posible

estandarizar y simplificar el proceso de conexión de aplicaciones Java a sistemas de gestión de bases de datos relacionales externos [JDBC, 2019]. Adicionalmente se emplea Python y el componente *cx_oracle*, el cual es un módulo de extensión de Python que permite el acceso a la base de datos Oracle [Cx_Oracle, 2019]. La conexión para el manejo de la información hacia la base de datos se realiza mediante el módulo de Python *commands*. Este toma un comando del sistema como una cadena y devuelve cualquier salida generada por el comando y, opcionalmente, el estado de la misma [Commands, 2019].

Control de asistencias en escritorio

El módulo perteneciente a la aplicación de escritorio se encuentra desarrollado en Python 2.7.16. La interfaz gráfica de usuario (GUI) está desarrollada con el módulo Tkinter, para Python 2.x. El uso de la biblioteca Tkinter permite crear cada una de las ventanas correspondientes a los requerimientos y funcionalidad de la aplicación [Tkinter, 2019].

Toma de asistencia (Raspberry)

El módulo para la toma de asistencia montado desde la tarjeta Raspberry Pi cuenta con una pantalla LCD 16x2, un lector biométrico Digital Persona U ARE U 4500, un led rojo, un led verde y un botón de verificación (figura 4).

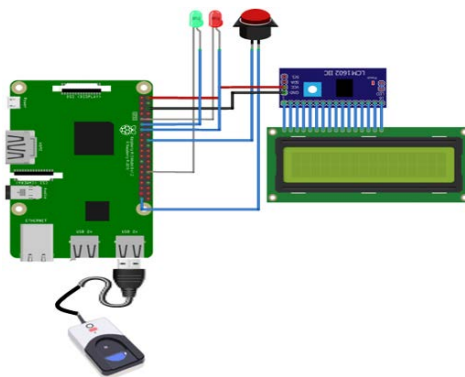


Figura 4 Conexión de componentes que conforman el módulo de toma de asistencias.

La funcionalidad programada en Python incluye las consideraciones de manipulación de una determinada tarea que permita al usuario tenerlo en un

contexto de los posibles errores y advertencias. Estos sucesos incluyen la conexión hacia la base de datos, desconexión del lector de huellas digitales, invalidez de los registros de asistencias, entre otros. Así se puede mostrar en la pantalla lcd un mensaje del pase de lista dado por el lector dactilar.

El sistema montado en el prototipo es autónomo. Esto quiere decir que al encender la tarjeta Raspberry Pi comenzarán a ejecutarse los procesos de inicialización (el flujo se puede ver en la figura 5). Este proceso se realiza mediante *Crontab*, el cual es un demonio en Linux que realiza un proceso de larga duración y que ejecuta comandos en fechas y horas específicas. Para este módulo se ocupan programar dos scripts; i) el primero permite tomar asistencia, y que se estará ejecutando cada vez que la tarjeta Raspberry Pi encienda; y ii) el segundo se usará para estar consultando y almacenando en el sistema los *templates* de huellas digitales de los alumnos en los grupos y las clases.

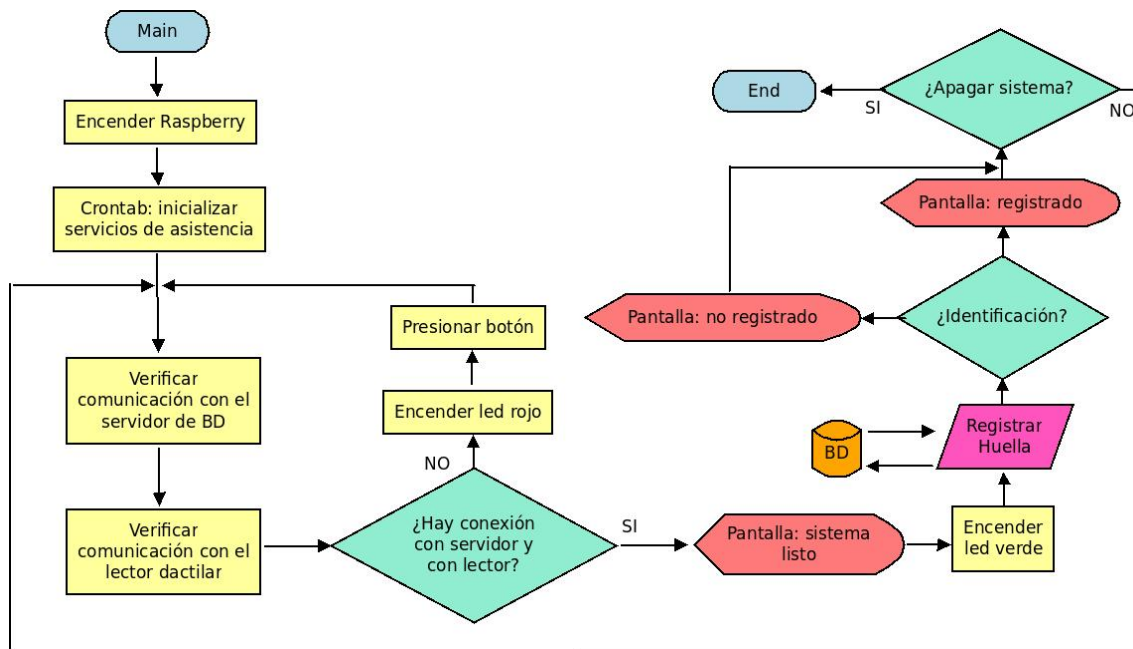


Figura 5 Descripción del programa funcionando con la Raspberry.

Al iniciar el programa dentro de la tarjeta de la Raspberry Pi, lo primero que hace el módulo desarrollado es verificar que exista comunicación con el servidor de la base de datos. Eso se hace por medio de una prueba de conectividad con el comando

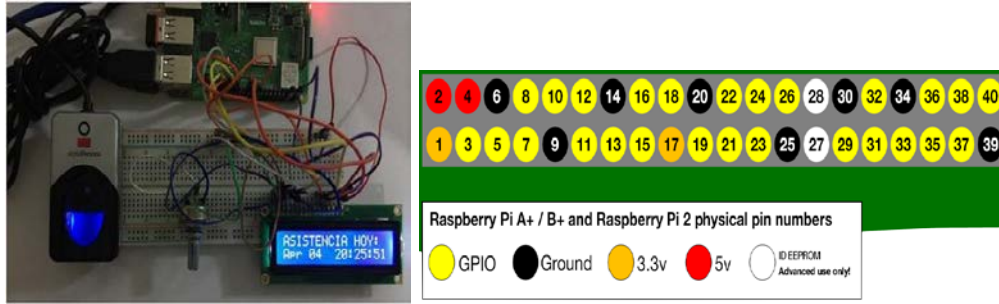
ping. Esta acción se integra conjuntamente con una consulta de la versión del servidor de base de datos con el fin de probar el acceso a ella. Esto se hace desde Python con la librería *subprocess*, el cual es un módulo que genera procesos que permiten conectarse a sus tuberías de entrada/salida/error y obtener sus códigos de retorno. Entonces si la comunicación con la base de datos fue exitosa, aparecerá en la pantalla lcd un mensaje que indica que el módulo está en línea con la base de datos. Para notificar de manera complementaria, aparecerá un led verde que indica el funcionamiento correcto del módulo para la toma de asistencias. En caso de que se suscite un error, se encenderá un led rojo y mostrará en la pantalla lcd el mensaje de error ocasionado. El segundo punto a checar es la corroboración del funcionamiento del lector biométrico Digital Persona.

Cuando las dos pruebas anteriores cumplen con las condiciones dadas, se procede a identificar al estudiante con su huella. Una vez que se tienen las huellas de alumnos enrolados por clase, el estudiante llega y coloca su dedo con el que fue registrado en un inicio. Entonces el programa identificará de qué alumno se trata, mostrando en la pantalla lcd la matricula, la fecha y hora a la clase que ingresó o abandonó. Si el alumno no se encuentre en la clase, se procede a mostrar un mensaje de error que notifica que debe intentarlo nuevamente.

Otros componentes a destacar en el sistema incluyen:

- Configuración del lector. Para capturar la asistencia del estudiante es necesario que exista compatibilidad de los *templates* generados desde el panel de control de la aplicación de escritorio; es por ello que las librerías usadas son las mismas, *libfprint* y *fprintd*. Conjuntamente, también se requiere el componente *Pyfprint*, el cual tiene como objetivo identificar al estudiante para agregar a la base de datos la asistencia correspondiente.
- Configuración de la pantalla LCD. La configuración de la pantalla lcd 16x2 (figura 6a), que se ocupa para mostrar información de avisos y notificaciones, fue empleada mediante el módulo I²C [Basics, 2019]. El protocolo de circuito Inter integrado (I²C) es un protocolo destinado a permitir que múltiples circuitos integrados digitales "esclavos" (chips) se comuniquen con uno o más chips "maestros" [Horne, 2019]. Posteriormente se requiere instalar la

librería *lcd_driver*, la cual contiene la programación para que se puedan mandar mensajes a la lcd mediante Python.



a) Módulo sistema biométrico

b) GPIO Raspberry Pi A+, B+ y Pi 2

Figura 6 Módulo biométrico y GPIO de modelos de tarjeta Raspberry Pi 2.

- Configuración de leds y botón. General Purpose Input Output (GPIO) es un sistema de entrada y salida de propósito general. GPIO consta de una serie de pines o conexiones que se pueden usar como entradas o salidas para múltiples fines. Estos pines están incluidos en todos los modelos de Raspberry Pi, aunque con ciertas diferencias [GPIO, 2019]. Para usar los GPIO es necesaria la librería RPi.GPIO de Python [RPi.GPIO, 2019]. Para saber cuáles GPIO usar es necesario tener en cuenta el diagrama que se expone en la figura 5 (b). Para el sistema se usaron tres GPIO:
 - ✓ El GPIO configurado como salida para el led verde (13) advierte al usuario que el sistema está trabajando correctamente.
 - ✓ El GPIO configurado como salida para el led rojo (27) advierte que el sistema ha fallado con la conexión de la base de datos o el lector de huellas digitales ha dejado de funcionar.
 - ✓ El GPIO configurado como entrada para el led rojo (18) reinicia el sistema si este no se restablece en un máximo de 10 reintentos.

Evaluar el Prototipo

En esta fase de evaluación del prototipo se dictamina qué elementos se pueden considerar aceptables y qué elementos se pueden considerar para ser sometidos a una nueva iteración de acuerdo al modelo de *Prototyping*. En el sistema

desarrollado se estuvieron haciendo varias pruebas en los diversos módulos. El subsistema que presentó más complicaciones y que tuvo que ser sujeto a diversas iteraciones fue el que realiza la toma de asistencia. Este subsistema sufría de varias incompatibilidades en los *templates* de las huellas digitales. Se hicieron diversas pruebas sobre el uso de Java y Python. En este mismo sentido se modificó la interfaz de usuario original del módulo de control de asistencias y reportes. Este subsistema fue modificado para ir agregando los diversos componentes de reportes a través de las interfaces de usuario de Tkinter. Finalmente, después de varias iteraciones, el prototipo ha alcanzado un nivel de madurez con una funcionalidad mostrada en los resultados del presente trabajo.

3. Resultados

Panel de Control de la Aplicación de Escritorio

En la figura 6 se muestra el panel de control principal de la aplicación de escritorio para el usuario (docente), la cual está dividida en tres secciones. La primera sección (parte superior de la ventana) está compuesta por cuatro botones de funcionalidad y una etiqueta para mostrar la referencia la fecha actual. El botón “Agregar Alumno” tiene la tarea de mostrar una ventana con el formulario correspondiente a sus datos. El segundo botón “Clase-Alumno” consiste en enrolar a los alumnos registrados a sus respectivas clases. El tercer botón “Generar Reportes” permite realizar los reportes de asistencia de forma individual, o si se requiere, en forma de clase-grupal. Por último, se tiene un cuarto botón que permite configurar la base de datos. Este último tiene la finalidad de generar un archivo de configuración para la base de datos, cuyo archivo es utilizado por el módulo de asistencia montado en la tarjeta Raspberry Pi.

En la sección 2 (parte izquierda de la ventana de la figura 7) se muestra un listado de los alumnos registrados en el sistema. Al dar un doble clic en alguno, se desplegará una ventana que contiene una serie de *widgets* que realizan dos acciones: editar o eliminar datos. En la sección 3 (parte derecha de la ventana de la figura 7) se tiene un panel para búsqueda y ordenamiento según sea el caso.

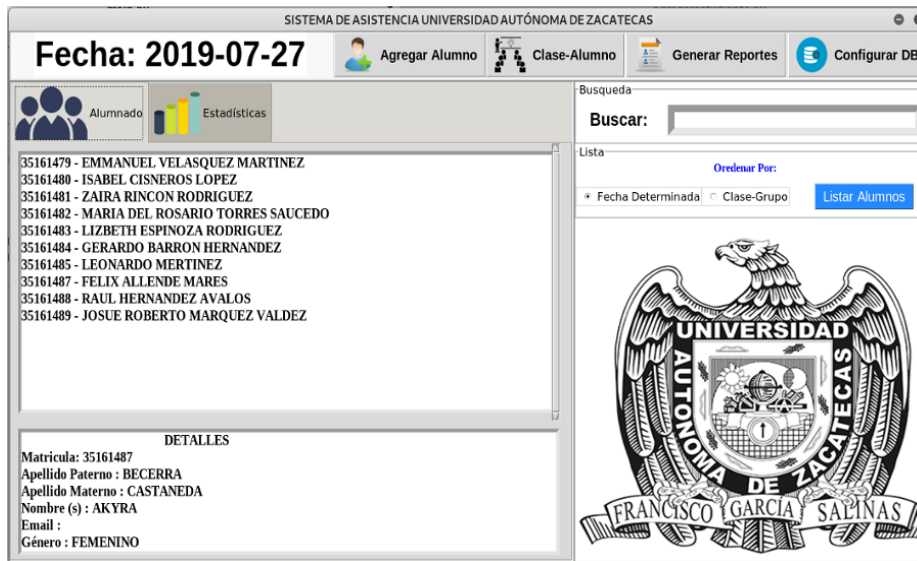


Figura 7 Panel del control de la aplicación de escritorio.

En la búsqueda se puede localizar al estudiante escribiendo la matricula o nombre y la sección 2 empezará a mostrar coincidencias. Adicionalmente se pueden ordenar los estudiantes según sea el caso, una fecha fija o por clase-grupo.

En la ventana mostrada en la figura 8 se tiene un formulario que brinda la funcionalidad de registrar los datos de los estudiantes: matricula, nombre, género, email y la digitalización de la huella dactilar.

Figura 8 Formulario para el registro de estudiantes.

Cada campo está validado usando expresiones regulares mediante Python. Estas funciones permiten verificar si una cadena particular coincide con una expresión regular dada [Regular expression, 2019]. Aquí se validó que la matrícula fuera un número, que los nombre(s) y apellido(s) tengan solamente letras, que el email sea válido, etc. Para capturar la huella es necesario usar elementos como un lector de huellas digitales, así como el proyecto fprint. Para la toma de huella digital del estudiante se implementó el uso de las librerías libfprint (habla con dispositivos de lectura de huellas digitales y procesar datos de huellas digitales) y fprintd (proporciona la funcionalidad de escaneo de huellas digitales a través de D-Bus) [Finger print, 2019].

Otro elemento usado para la captura de la huella dactilar es el módulo Pyfprint. Este posee varios componentes que permiten digitalizar y verificar la huella capturada, además de poder identificar una huella específica [Pyfprint, 2019]. El módulo Pyfprint necesita que la persona a registrar coloque un total de cinco veces su dedo. Posteriormente se realizan dos verificaciones de la huella.

Para generar un reporte de asistencia de un alumno es necesario seleccionar el alumno y presionar el botón de generar reportes. El reporte está basado en *ReportLab*, que es un motor de código abierto, robusto y que permite crear documentos PDF complejos y basados en datos y gráficos vectoriales personalizados [ReportLab, 2019]. El reporte generado en la figura 9 muestra el nombre completo del alumno, Programa Académico al que está inscrito, la matrícula, la materia de la clase, el nombre del docente, el ciclo escolar, el historial de la clase, las clases totales, las clases asistidas por el alumno, el total de retardos (3 retardos equivalen una falta), las faltas totales y finalmente, el porcentaje total del alumno, incluyendo la fecha de expedición del reporte.

Módulo de Registro de Asistencias

En la figura 10a se observa el estado del módulo de toma de asistencia. En la pantalla lcd 16x2 se muestra un mensaje (juntamente con el led verde) que indica que el módulo se encuentra funcionando correctamente (respecto a la conexión a la base de datos y al lector biométrico). En la figura 10b se evidencia como un

alumno predeterminado llega y coloca su dedo con el que se cotejará al registrado en un inicio. Para este paso se recomienda dejar el dedo que trae consigo la huella dactilar en el lector hasta que se arroje un mensaje en la lcd con su matrícula, fecha y hora; o en caso contrario un mensaje de error.

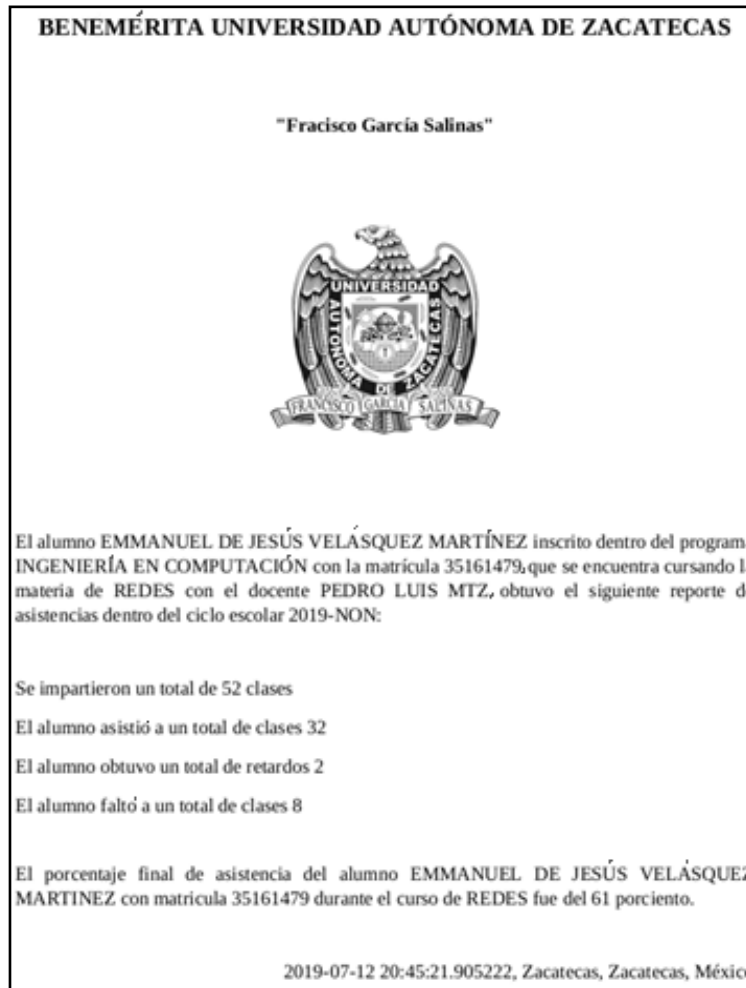


Figura 9 Reporte generado por Reportlab para las asistencias de un alumno a clase.



a) En Línea

b) Disponible para lectura

Figura 10 Estado de en línea para toma de asistencia en módulo biométrico.

En la figura 11a se revela el pase de lista del estudiante. El mensaje que arroja en la pantalla lcd es la matricula del alumno, así como la fecha y hora de entrada (figura 11b), registrando esta información al instante en la base de datos.



Figura 11 Toma de asistencia y mensaje en donde se encontró el alumno inscrito.

En caso de que se muestre el mensaje “INTENTE DE NUEVO” (figura 12), se puede deber a dos casos: i) el estudiante no se encuentra inscrito en el curso, ii) la huella dactilar de la persona se encuentra sucia. En las figuras 12a y 12b se muestran dos posibles errores durante el uso del módulo de la toma de asistencia: i) el lector no está conectado o se encuentra dañado; ii) no se puede establecer la comunicación hacia la base de datos, reintentando restablecer el enlace.



Figura 12 Mensaje de error indicando no coincidencias.



Figura 13 Casos de error en el módulo biométrico.

4. Discusión

Las pruebas generales y el uso parcial que se le ha dado al sistema de asistencias han mostrado resultados favorables. Sin embargo, paulatinamente se le han ido diseñando nuevas validaciones que ayudan a mejorar su funcionalidad. Por ejemplo, se le han ido plasmando consideraciones de altas y registros de profesores para que pueda brindar movilidad y cambio de estafeta del prototipo de un docente a otro.

De manera complementaria, se le han ido diseñando validaciones de horas de entrada con respecto a diversos grupos en horarios determinados.

También se le han estado haciendo adecuaciones para incluir reportes que el alumno puede obtener de su propio historial de asistencias en el semestre, esto de manera automática a través de un servidor SMTP. Estas adecuaciones estéticas y de funcionalidad se han ido detectando por las necesidades y sugerencias de los involucrados en los resultados de las pruebas que se le ha ido dando al sistema en un contexto real. Se puede considerar que el sistema es económico, funcional y de escalabilidad sin límites. Bajo el marco de trabajo de la puesta en marcha del prototipo, se debe recordar y tomar en cuenta la probabilidad y consideraciones de cambios futuros en cualquier diseño y desarrollo de software [Becerra, 2018], [Weitzenfeld, 2007]. Las “interfaces” representan los elementos gráficos, la “funcionalidad” son los requisitos del usuario, los “datos” y “funciones” equivalen a los componentes internos que se usan para describir a los “objetos”, en tanto que la “información” representa el dominio del problema en una aplicación. Cabe resaltar que la arquitectura del sistema presentado debe distinguir entre elementos con mayor y menor probabilidad de cambios.

De igual manera, el desarrollo del sistema de software propuesto debe contemplar un modelo de procesos en el que los componentes de mayor probabilidad de cambio no “arrastren” a los más estables.

Las herramientas funcionales del prototipo propuesto han propiciado un cambio constante en su diseño para adecuarse a los momentos cambiantes del enfoque educativo.

5. Conclusiones

En el presente trabajo se diseñó y desarrolló un prototipo para el registro de asistencias de alumnos en un ambiente universitario. El tener implementado un sistema de control de este tipo dentro del aula de una institución educativa de nivel superior ya ha permitido obtener un informe de asistencia de los alumnos. Este tipo de infraestructura ha coadyuvado en ofrecer al profesor un apoyo para combatir el desempeño de los estudiantes. De esta manera se ha obtenido más preciso el historial a través de los grupos de prueba de los estudiantes, obteniendo estadísticas y apoyo en proceso de tutoría. El funcionamiento del sistema ha evidenciado que los tiempos consumidos en los procesos manuales de asistencia pueden verse favorecidos en gran medida con procesos automáticos y de gran eficiencia, brindando rapidez en su control. No obstante, el cambio en el sistema se ha ido dando y se tendrá que adecuar a constantes adecuaciones a las necesidades sugeridas por los usuarios que son sujetos en las pruebas.

Se espera que en poco tiempo el sistema sufra un crecimiento de uso y aceptación en la comunidad. Para trabajo futuro se plantea que el sistema permita migrar a nuevos tipos de mecanismos biométricos, como reconocimiento de voz.

6. Bibliografía y Referencias

- [1] Basics, C. (2019). How to Setup an I2C LCD on the Raspberry Pi – Circuit Basics, consultada el 27 de julio de 2019: <http://www.circuitbasics.com/raspberry-pi-i2c-lcd-set-up-and-programming/>.
- [2] Becerra, A. (2018). Aplicación gestora de contenidos de aprendizaje personalizados para un ambiente híbrido de tipo virtual-presencial, *Academia Journals Morelia*.
- [3] Benavides, G, Sosa, C., & Montané, L. (2016). Reconocimiento facial aplicado a la toma de asistencia. *Pistas Educativas*, No. 122, 57-68.
- [4] Calle, V. (2016). Control de asistencia a clase mediante un lector de huella digital (tesis de grado), Mérida, Yucatán.
- [5] Commands. (2019). Commands - Utilities for running commands: <https://docs.python.org/2/library/commands.html>.

- [6] CRUD. (2019). What is CRUD? [https://www.codecademy.com/articles /what-is-crud](https://www.codecademy.com/articles/what-is-crud).
- [7] Cx_Oracle. (2019). cx_Oracle - Python Interface for Oracle Database: https://oracle.github.io/python-cx_Oracle/.
- [8] Finger print. (2019). Finger print reader support: <https://fprint.freedesktop.org/>.
- [9] GPIO. (2019). GPIO (General Purpose Input/Output) Definition: <https://techterms.com/definition/gpio>.
- [10] Horne, M. (2019). I2C 20x4 LCD character display on a #RaspberryPi: <https://www.recantha.co.uk/blog/?p=4849>.
- [11] JDBC. (2019). What is Java Database Connectivity (JDBC)? <https://www.theserverside.com/definition/Java-Database-Connectivity-JDBC>
- [12] Jiménez, G. (2018). Sistema web de control de asistencia basado en web services y la biometría de huella dactilar para las instituciones educativas (tesis de grado), Universidad Nacional Mayor de San Marcos, Lima, Perú.
- [13] Márquez, L. E., Abdo, Y. N., & Ángulo, F. J. (2014). Prototipo de control de acceso a aulas y registro automático de asistencia. *Revista colombiana de Tecnologías Avanzadas*, Vol 2, No. 26, 41-47.
- [14] Moodle. (2019). Moodle: <https://moodle.org/?lang=es>.
- [15] Oracle Database. (2019). Introduction to the Oracle Database: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm.
- [16] Pyfprint. (2019), Pyfprint: [//github.com/luksan/pyfprint/tree/master/pyfprint](https://github.com/luksan/pyfprint/tree/master/pyfprint).
- [17] Python. (2019). What is Python? Executive Summary: <https://www.python.org/doc/essays/blurb/>.
- [18] Regular expression (2019). Regular expresión operations: <https://docs.python.org/2/library/re.html>.
- [19] ReportLab. (2019). Open Source - ReportLab.com: <https://www.reportlab.com/opensource/>.
- [20] Rouse, M. (2005). What is Prototyping Model? <https://searchcio.techtarget.com/definition/Prototyping-Model>.
- [21] RPi.GPIO. (2019). RPi.GPIO: <https://pypi.org/project/RPi.GPIO/>.

- [22] Sepúlveda, L., & Coavas, Y. (2014). Desarrollar una plataforma web para el control de asistencia de estudiantes, docentes y administrativos de la universidad de córdoba sede lorica mediante el uso de tarjetas inteligentes, dispositivos móviles y SMS (tesis de grado), Lorica, Córdoba.
- [23] Sommerville, I. (2001). *Software engineering*. Boston: Addison-Wesley.
- [24] Tkinter. (2019). Tkinter - Python interface to Tcl/Tk: <https://docs.python.org/2/library/tkinter.html>.
- [25] Weitzenfeld, A. & Guardati, S. (2007). Capítulo 12: Ingeniería de software: el proceso para el desarrollo de software. Libro: *Introducción a la Computación*. México D.F: CENGAGE Learning.