

UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

---



FACULTAD DE INGENIERÍA

TESINA

**SISTEMA DE CONTROL DE ENTRADA  
Y SALIDA POR MEDIO DE CÓDIGO  
QR PARA LA DECDFI.**

QUE PARA OBTENER EL TÍTULO  
DE INGENIERO EN COMPUTACIÓN

PRESENTA:

**Román Guadarrama José Roque**

DIRECTOR DE TESINA:

**Luis Sergio Valencia Castro**

CIUDAD UNIVERSITARIA FEBRERO DE 2013



## Contenido

Índice de contenido	3
Introducción	4
Proyectos elaborados para la DECD	5
1. Marco Teórico	7
1.1. OpenCV	7
1.2. QR code	9
1.3. SQLAPI Library (Free trial)	12
1.4. Libdecodeqr	12
1.5. Desarrollo ágil de software como metodología de desarrollo	12
2. Desarrollo del proyecto: Credenciales de control de acceso de entrada y salida con Código QR	15
2.1. Planteamiento inicial	15
2.1.1. Diagrama de flujo inicial.	15
2.1.2. Diagrama de flujo final.	16
2.1.3. Creación de los códigos QR.	17
2.2. Metodología de Ingeniería de Software empleada	18
2.3. Implementación de la Base de Datos.	19
2.4. Código de la aplicación en lenguaje C++.	20
2.5. Descripción de los pasos que ejecuta el programa en su ejecución.	22
2.6. Diseño de las credenciales.	26
3. Resultados obtenidos del proyecto	27
4. Conclusiones	28
4.1. Proyecciones a Futuro	29
5. Apéndices	30
5.1. Apéndice A	30
5.2. Apéndice B	32
5.3. Apéndice C	34
6. Glosario	44
7. Referencias	45

## Índice de contenido

### Índice de tabla

Tabla 1.1 Sistemas Operativos donde se puede instalar OpenCV ..... 8 Tabla

Tabla 1.2 Tamaño de los módulos del código QR de acuerdo a la  
resolución de las impresoras comerciales ..... 11

### Índice de imágenes

Imagen 2-1. Diagrama de flujo inicial de la BD ..... 15

Imagen 2-1. Diagrama de flujo final de la BD ..... 16

Imagen 2-3. Códigos QR generados ..... 17

Imagen 2-4. Diagrama de relación inicial de la BD ..... 19

Imagen 2-5. Diagrama de relación final de la BD ..... 20

Imagen 2-6. Pantalla de conexión a la BD ..... 24

Imagen 2-7. Pantalla de las instrucciones del programa ..... 25

Imagen 2-8. Pantalla de detección de código incorrecto ..... 25

Imagen 2-9. Pantalla de detección de código que no es el  
de la DECDFI ..... 26

Imagen 2-10 Pantalla de detección de código que es el  
de la DECDFI ..... 26

Imagen 2-11 Diseño frontal de la credencial ..... 27

Imagen 2-12. Diseño trasero de la credencial ..... 27

## Introducción

En julio de 1971 la Facultad de Ingeniería de la UNAM decidió crear el Centro de Educación Continua, dependiente de la División de Estudios Superiores, con el fin de poder proporcionar a los egresados de la Facultad de Ingeniería y de símiles, un medio con el cual poder actualizar los conocimientos adquiridos durante la etapa estudiantil, y de esta forma poder estar al día con las necesidades requeridas en el mundo laboral, puesto que con la rápida evolución de la ciencia y la tecnología el ingeniero tiene que estar actualizándose constantemente.

Actualmente el Centro de Educación Continua figura bajo el nombre de “División de Educación Continua y a Distancia (DECD)” se encuentra dentro del Palacio de Minería (ubicado en Tacuba N° 5, en el Centro Histórico del D.F) y entre las actividades que imparte, se encuentran: los cursos de actualización profesional para ingenieros; los cursos de adiestramiento en materias específicas y los cursos patrocinados por dependencias gubernamentales para mejorar el desempeño de su personal técnico.<sup>1</sup>

Dentro de ésta, una de sus más recientes áreas es la de “Investigación y Desarrollo (I+D)”, la cual tiene como finalidad investigar, desarrollar e implementar aplicaciones novedosas teniendo como base las nuevas tecnologías y herramientas disponibles y para lo cual está organizada en varias ramas.

Una de estas ramas de trabajo que se tienen en el área de Investigación y Desarrollo, es la de “Realidad Aumentada y Visión Computacional”; en esta área de especialización es donde se diseñaron, desarrollaron e implementaron aplicaciones que hacen uso de técnicas tecnológicas innovadoras para resolver problemas cotidianos del quehacer humano.

En esta tesina se presenta el desarrollo del proyecto elaborado para el área de Investigación y Desarrollo, que tuvo como finalidad una aplicación funcional que cumpliera con los requerimientos planteados en la etapa de diseño y que se logró aplicando los conocimientos adquiridos durante las clases curriculares en la Facultad de Ingeniería, así como aprendiendo más acerca de tecnologías y herramientas enfocadas al área de Visión Computacional y Computación Gráfica.

## Proyecto elaborado para la DECD

El proyecto a continuación presentado, se elaboró en base a las necesidades que se tenían en la DECD, siendo una de éstas la de poder tener un registro de las personas que laboran para dicha área. La necesidad de este registro se debía a que dentro ella se cuenta con 6 tipos de colaboradores, los cuales tienen diferentes horarios y proyectos asignados. Por lo tanto, para resolver esta problemática, se planteó la necesidad de implementar un sistema de chequeo de entrada y salida.

Los 6 tipos de colaboradores existentes dentro del área son:

1. Becario: Con 4 horas diarias. Puede reponer horas otro día si se falta previo aviso. Recibe un pago mensual.
2. Honorarios: Con 8 Horas diarias. Se pueden reponer horas otro día si se falta previo aviso. Recibe un pago mensual.
3. Tesista: Con 2 horas diarias. Se puede faltar previo aviso. No es necesario reponer horas. No recibe pago.
4. Base: Con 8 horas diarias. No se puede faltar, de ser así se descuenta el pago de horas si no se justifica la falta. Recibe un pago quincenal.
5. Servicio Social. Con 4 horas diarias. Se deben de reponer horas y justificar la falta. No recibe pago.
6. Prácticas Profesionales: Con 4 horas diarias. Se deben de reponer horas y justificar la falta. No recibe pago,

De acuerdo a estas características de los distintos tipos de colaboradores, se buscaba como finalidad tener un control de horas presenciales diarias dentro del área.

Para poder elaborar una aplicación adecuada a esta problemática, el punto de partida que se tomó fue el de saber que se manejaría una cámara web, por lo tanto las dos librerías que manejan dicho dispositivo y que son más aptas son ARToolkit y OpenCV.

Estas dos librerías manejan las imágenes que se reciben por la webcam; sin embargo, OpenCV está más enfocado a la manipulación de las imágenes, mientras que ARToolkit se enfoca más al reconocimiento de patrones. Debido a este planteamiento, se pensó que ARToolkit serviría más al propósito del sistema; pero esta librería no tiene soporte oficial desde hace un par de años y no está diseñada para sistemas operativos de 64 bits, por lo tanto la librería OpenCV fue la elegida, esto

aunado a que ya se había utilizado anteriormente para otro proyecto.

Así mismo, se sabía que se tendría que usar una base de datos para poder llevar el registro de las entradas y salidas y así poder tener algún identificador para cada una de las personas que laboraban.

Teniendo esta información a la mano surgió la incógnita de cómo poder relacionar el identificador a una imagen que pudiera ser manipulada por medio de la librería OpenCV; de esta interrogante es de donde se obtuvo la idea de usar el código QR, el cual ya se había visto en uso en algunas páginas web y algunos medios de comunicación de internet así como su uso en otros países.

# 1. Marco Teórico

En este capítulo se presenta una breve introducción acerca de cada una de las herramientas y tecnologías que se usaron para el desarrollo del proyecto requerido dentro del área de I+D.

## 1.1 OpenCV

OpenCV (Open Source Computer Vision Library) es una librería de visión computacional desarrollada bajo licencia de código libre, está programada en los lenguajes C y C++ y es independiente del sistema operativo, del hardware y de la interfaz de ventanas; por lo tanto, puede ser instalada en sistemas Linux, Mac y Windows. También se planea su integración con los lenguajes Python, Ruby, Matlab entre otros.

Su desarrollo principalmente fue enfocado para dar una mayor eficiencia y optimización en uso de aplicaciones en tiempo real, esto sobre todo gracias a que está programado bajo lenguaje C optimizado que permite el uso de procesadores de múltiples núcleos. Otra característica con la cual se desarrolló es que se puede integrar junto con las librerías IPP (Integrated Performance Primitives) de Intel para mejor eficiencia. Es una API tanto de bajo como de alto nivel y permite una manipulación genérica tanto a imágenes como a video.

Entre las 500 funciones integradas que tiene, se pueden desarrollar aplicaciones para las áreas de procesamiento de imágenes, imágenes médicas, seguridad, interfaces de usuarios, visión estéreo y como el procesamiento de imágenes la mayoría de las veces se liga con el área de la mecánica, en especial la robótica, incluye una librería de propósito general para aprendizaje de máquinas (MLL), enfocada sobre todo para reconocimiento estadístico de patrones y clústeres.

Con las funciones incluidas en la librería, se puede hacer manipulación de los datos de imágenes, entrada y salida de imágenes y video, manipulación de vectores y matrices, rutinas de álgebra lineal, procesamiento de imágenes básico( filtros, detección de bordes, región de interés (ROI), muestreo e interpolación, conversión a diferentes escalas de colores, histogramas), calibración de cámara, análisis de movimiento, reconocimiento de objetos y desarrollo de interfaces de usuario(GUI)

2  
simples.

Los 4 módulos principales que conforman esta librería son:

- cv- Las funciones principales de OpenCV
- cvaux- Funciones auxiliares y experimentales de OpenCV
- cxcore- soporte para álgebra lineal y estructuras de datos
- hughgui- funciones de GUI

Esta librería estuvo en desarrollo desde 1999 hasta el año 2007 en el que fue lanzada la versión 1.0. Su desarrollo empezó como una iniciativa del departamento de investigaciones de Intel, se buscaba una forma en que se desarrollarían aplicaciones con alto costo computacional. Se hizo un sondeo en varias universidades y se recopiló funciones y frameworks desarrolladas por dichas universidades, entre ellas el MIT. Una vez que se tuvieron todas estas aplicaciones, se envió a la división de Intel en Rusia, en donde Vadim Pisarevsky manipuló, codificó, optimizó y programó la mayor parte de OpenCV, con la ayuda de Victor Eruhimov y Valery Kuriakin pudo desarrollar la infraestructura y seguir adelante con el proyecto.

Uno de los objetivos principales que se tenían al desarrollarlo, era que se tuviera un avance importante en el campo de la visión, permitiendo que se pudiera programar más fácilmente y de forma más entendible lo relacionado con el campo de la visión computacional, así como que se pudieran desarrollar aplicaciones comerciales con código portable y libre.

Intel hizo esto porque significaría más requerimiento de uso de procesador, y de esta forma obtendrían más ventas de sus procesadores multinúcleo lo que finalmente fue así y les permitió que el costo de desarrollo de este proyecto se hiciera auto- sostenible.

Durante su desarrollo hubo 5 versiones beta antes de la versión 1.0 y desde la primer versión beta (2001) se incluyó el soporte para sistemas Linux. Y si bien fue Intel quien empezó este proyecto, no es el autor ni propietario de él, puesto que lo publicó como código libre.

En la tabla 1.1 muestro los diferentes OS y arquitecturas de computadoras donde se puede instalar OpenCV.



	IA32	EM64T	IA64	Other (PPC, Sparc)
Windows	✓ (w. IPP; MSVC6, .NET2005+OMP, ICC, GCC, BCC)	✓ (w. IPP; MSVC6+PSDK, NE T2005+OMP, PSDK)	± (w. IPP; PSDK, some tests fail)	N/A
Linux	✓ (w. IPP; GCC, BCC)	✓ (w. IPP; GCC, BCC)	✓ (GCC, ICC)	✗
MacOSX	✓ (w. IPP, GCC, native APIs)	? (not tested)	N/A	✓ (iMac G5, GCC, native APIs)
Others (BSD, Solaris...)	✗	✗	✗	Reported to build on UltraSparc Solaris

*portability guide for release 1.0: operating systems are shown on the left; computer architecture types across top*

**Tabla 1.1 Sistemas Operativos donde se puede instalar OpenCV**

La librería se puede descargar desde este link: <http://SourceForge.net/projects/opencvlibrary> y ahí mismo se encuentra el wiki con las instrucciones necesarias para instalarlo en los diferentes OS (<http://opencvlibrary.SourceForge.net>) <sup>3</sup>

## 1.2 QR Code

### Introducción

Los QR Code o códigos de barra bidimensionales fueron creados en 1994 por la compañía japonesa Denso-Wave con la finalidad de tener un código de barras que sea fácilmente leído por los scanners y que además pueda almacenar más información que un código de barras unidimensional.

### Características

Las características que significan una mejora con respecto al código de barras tradicional son:

- Almacenar la misma cantidad de información que un código de barras tradicional en un 10% menor de impresión.
- Capacidad de corrección por daño con un nivel de recuperación de hasta 30% del código en caso de manchas o daños.
- Lectura de 360° gracias al uso de patrones de detección de posición.
- Apéndice Estructural, el cual permite que la información de un código pueda ser a su vez representada como la información contenida en varios subcódigos, teniendo como un límite máximo de 16.

## **Estandarización**

Este tipo de código está regularizado bajo las siguientes organizaciones de estándares:

- AIM International lo aprobó en octubre de 1997 bajo el nombre (ISS- QR Code).
- JEIDA lo aprobó en marzo de 1998 bajo el nombre de (JEIDA-55).
- JIS lo aprobó en enero de 1999 bajo el nombre de (JIS X 0510).
- ISO lo aprobó en enero de 2000 bajo el nombre de (ISO/IEC18004).
- JIS aprueba en noviembre de 2004 el micro QR Code bajo el nombre de (JIS X 0510).

## **Almacenamiento**

Los QR Code pueden almacenar diferentes tipos de datos con una cantidad máxima de:

- 7,089 caracteres de tipo numérico.
- 4,296 caracteres de tipo alfanumérico.
- 2,953 caracteres de tipo binario.
- 1,817 caracteres de tipo kanji, hiragana o kana.

Para la corrección de error se manejan 4 niveles:

- Nivel L: Aproximadamente 7% del contenido del código puede ser restaurado.
- Nivel M: Aproximadamente 15% del contenido del código puede ser restaurado.
- Nivel Q: Aproximadamente 25% del contenido del código puede ser restaurado.
- Nivel H: Aproximadamente 30% del contenido del código puede ser restaurado.

## **Versión**

De acuerdo a la cantidad de información almacenada los QR Code se clasifican en versiones, las cuales van desde la versión 1 hasta la 40; cada versión tiene un patrón de acomodo de módulos dentro del código, el número de módulos va desde 21x21 en la versión 1, hasta 177 x 177 módulos en la versión 40 (incrementando en cantidades de 4x4 módulos en cada versión).

## **Características del diseño**

Cuando se sabe a qué versión corresponde el QR que se va a generar, es importante tomar en cuenta como siguiente punto el tamaño que tendrá cada módulo (cada punto negro o blanco que formará el código) para poder tener un código que sea detectado por los escáneres y cámaras de los dispositivos móviles.

El primer valor a tomar en cuenta es que cada módulo debe estar conformado al menos por un arreglo matricial de 4x4 puntos, si se toma esta consideración, los diferentes tamaños por módulo de acuerdo a la resolución de las impresoras comerciales se ve en la tabla 1.2.

Impresora	Densidad de impresión	Módulo de 4x4 puntos	Módulo de 5x5 puntos	Módulo de 6x6 puntos
Láser	600 dpi	0.17 mm	0.21 mm	0.25 mm
Láser	360 dpi	0.28 mm	0.35 mm	0.42 mm
Tinta	300 dpi	0.33 mm	0.42 mm	0.5 mm
Tinta	200 dpi	0.5 mm	0.63 mm	0.75 mm

**Tabla 1.2 Tamaño de los módulos del código QR de acuerdo a la resolución de las impresoras comerciales**

En base a este dato, se debe de saber que generalmente los escáneres de alta resolución alcanzan un valor de 0.17 mm, mientras que los estándares manejan una resolución de 0.25 mm y en cuanto a la resolución de los dispositivos móviles, varía de dispositivo a dispositivo.

Es importante mencionar que el QR Code requiere tener un margen blanco de un tamaño de al menos 4 módulos a cada lado del símbolo.

Toda la información correspondiente a como se codifican los valores para generar el QR Code está especificada en el documento: ISO/IEC 18004 que tiene por título:

“Information technology — Automatic identification and data capture techniques — Bar code symbology — QR Code” [http://raidanii.net/files/datasheets/misc/qr\\_code.pdf](http://raidanii.net/files/datasheets/misc/qr_code.pdf)<sup>4</sup>

### **1.3 SQLAPI ++ Library (Free Trial)**

Esta librería fue usada por su compatibilidad con C++, además de que es compatible con la base de datos MySQL y de forma nativa tiene compatibilidad con el compilador de Microsoft Visual C++ y Visual Studio.

Como está desarrollada usando las API's de los diversos sistemas de gestión de bases de datos (o DBMS e inglés) permite una conexión directa con la base de datos sin requerir el uso de algún enlace e incrustación de objetos para base de datos (en inglés OLEDB), ni tampoco de una conectividad abierta de base de datos (en inglés ODBC).<sup>5</sup>

### **1.4 Libdecodeqr**

Librería de código libre, creada con la finalidad de decodificar QR Code bajo los estándares JIS x 0510 e ISO/IEC 18004. Desarrollada por Joseph Anthony Pasquale Holsten y enfocada a ser usada bajo el lenguaje de programación C++, requiriendo adicionalmente el uso de OpenCV.<sup>6</sup>

### **1.5 Desarrollo ágil de software como metodología de ingeniería de software**

Estas metodologías se enfocan en la premisa de poder tener un desarrollo iterativo e incremental, de tal forma que se tenga un producto funcional en poco tiempo, el cual posteriormente se puede ir modificando, corrigiendo y mejorando.

Estas metodologías tienen como base los 12 principios del manifiesto ágil que son <sup>7</sup> :

- 1 Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

- 2 Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- 3 Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- 4 Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 5 Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- 6 El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- 7 El software funcionando es la medida principal de progreso.
- 8 Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- 9 La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- 10 La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- 11 Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- 12 A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Entre las metodologías se lista Scrum, la cual se basa en iteraciones con una duración de no más de 4 semanas.

Los roles que existen son:

- Scrum Master: Encargado de ver que se cumplan las reglas.
- Product Owner: La voz del cliente, plantea requerimientos en el product backlog.
- Equipo de desarrollo: Responsables de entregar el proyecto, ver diseño, implementación, desarrollo, pruebas.

Los documentos que se manejan son:

- Product backlog: Lleva los requerimientos jerarquizados por retorno sobre la inversión. Ayuda a estimar la línea temporal.

- **Sprint backlog:** El equipo describe cómo va a implementar los requisitos en el siguiente sprint. Las tareas se dividen en horas de no mayor duración a 16 horas, estas tareas no son asignadas, sino que el equipo decide quien la toma.
- **Burn down:** Gráfica que mide la cantidad de tareas pendientes al inicio de cada Sprint, su tendencia es descendiente.

Las reuniones que se manejan son:

- **Daily Scrum:** Es cada día antes de empezar el sprint, dura no más de 15 minutos, es a la misma hora y en el mismo lugar siempre. Se plantean 3 preguntas: qué se hizo ayer, qué se va a hacer hoy y qué problemas se han encontrado.
- **Scrum de Scrum:** Es después del Daily Scrum, en ésta asiste 1 persona de cada equipo, sirve para que los equipos sepan cómo van los otros y se plantean las 4 preguntas: qué hizo tu equipo ayer, qué planean hacer hoy, han tenido dificultades y si se va a hacer algo que interactúe directamente con otro equipo.
- **Sprint Planning Meeting:** Al inicio de cada sprint se asignan las tareas a cubrir en el Sprint backlog y se comunica el estimado de trabajo que se realizará en este nuevo ciclo.
- **Sprint Review Meeting:** Reunión con el Product Owner al finalizar el sprint, se muestra un demo con el trabajo terminado y se comenta cual fue el trabajo que no se pudo completar.
- **Sprint Retrospective:** Al finalizar el Sprint, todos los miembros del equipo hacen una retrospectiva de lo que sucedió durante este ciclo con la finalidad de realizar una mejora continua del proceso.

## 2 Desarrollo del proyecto: Credenciales de control de acceso de entrada y salida con Código QR

### 2.1 Planteamiento inicial

El primer planteamiento que se propuso fue en base a la imagen 2-1, este diagrama de flujo inicial de la base de datos se hizo tomando en cuenta las herramientas que el análisis efectuado indicó que eran las más eficientes para el desarrollo de este proyecto.

#### Diagrama de flujo inicial

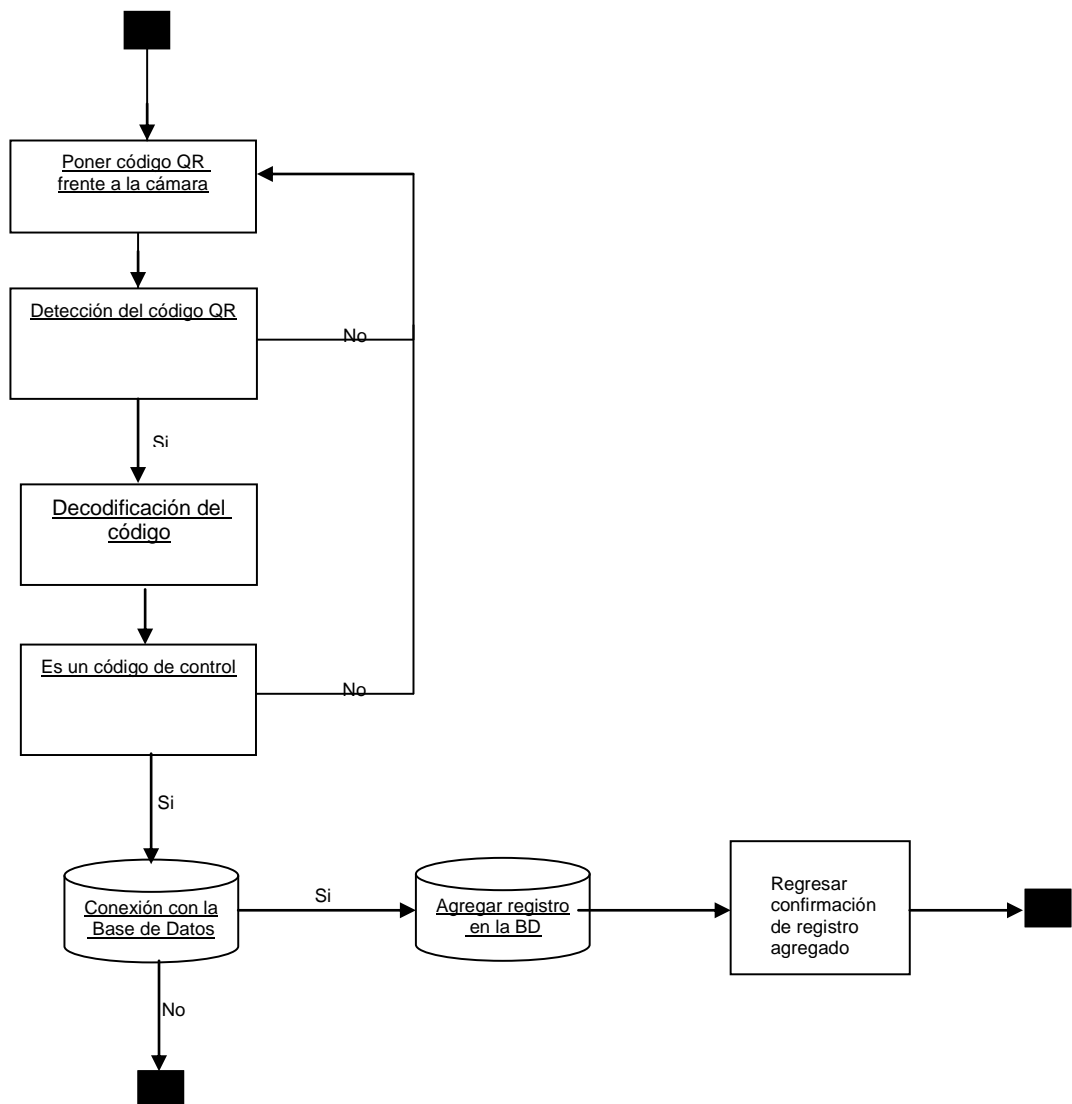
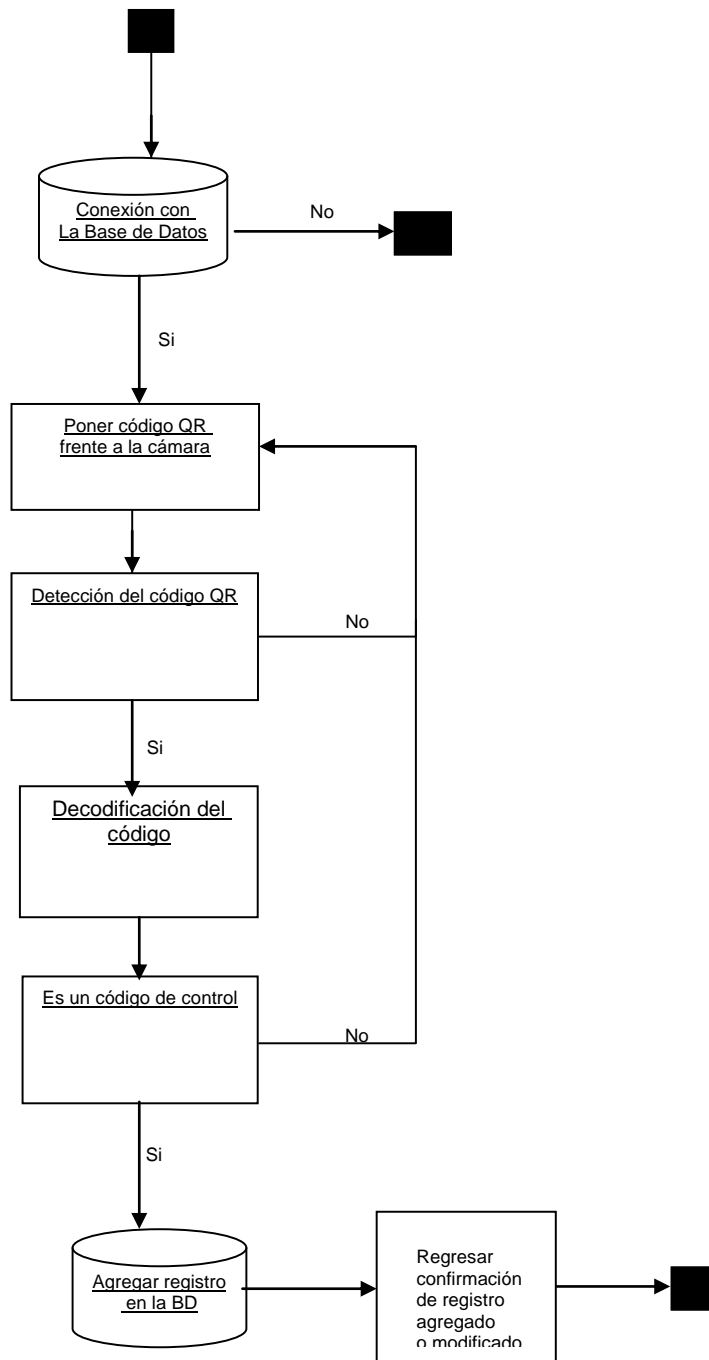


Imagen 2-1. Diagrama de flujo inicial de la BD.

En el proceso de la programación y de las pruebas cambió la forma en la cual funciona esta aplicación; sin embargo, este primer algoritmo planteado no sufrió modificaciones críticas.

La imagen 2-2 muestra el diagrama de flujo final de la base de datos.

### 2.1.1 Diagrama de flujo final



**Imagen 2-2.** Diagrama de flujo final de la BD.



## 2.1.1 Creación de los códigos QR

A pesar de que existe la norma ISO para la generación del código QR, es sabido que son una serie de reglas y pasos que dan pie a diferentes interpretaciones, sobre todo cuando quién genera el código QR lo hace en base a la sola lectura de la ISO; tal como lo expresa Mr. Balas, CEO de VitreoQR

“Entre la patente de DENSO, la cual se puede conseguir una copia por diez dólares y la copia del documento referente al ISO 18004 que puede ser comprada por 350 dólares, los programadores creen que tienen la receta entera para poder desarrollar su propio generador de código QR; cuando en realidad, se tienen los componentes básicos más no todas las instrucciones de ensamblado. Cada uno de los generadores de código QR que son de código libre ha sido construido en base a las ‘interpretaciones’ que alguien obtuvo de estos documentos. Con cada ensamblado sale un resultado diferente.”<sup>8</sup>

Teniendo en cuenta lo expresado anteriormente decidí usar los códigos generados en la página de <http://www.qrcode.es/> ; ya que de 15 códigos generados, solo 1 dio problemas para detectarse, mientras que los códigos producidos en otros generadores web disponibles dieron error en más de 4.

Entre los generadores web que descarté están:

- <http://qrcode.kaywa.com/>
- <http://www.qrstuff.com/>
- <http://zxing.appspot.com/generator/>
- <http://goqr.me/>

Algunos de los códigos que generé son los mostrados en la imagen 2-3



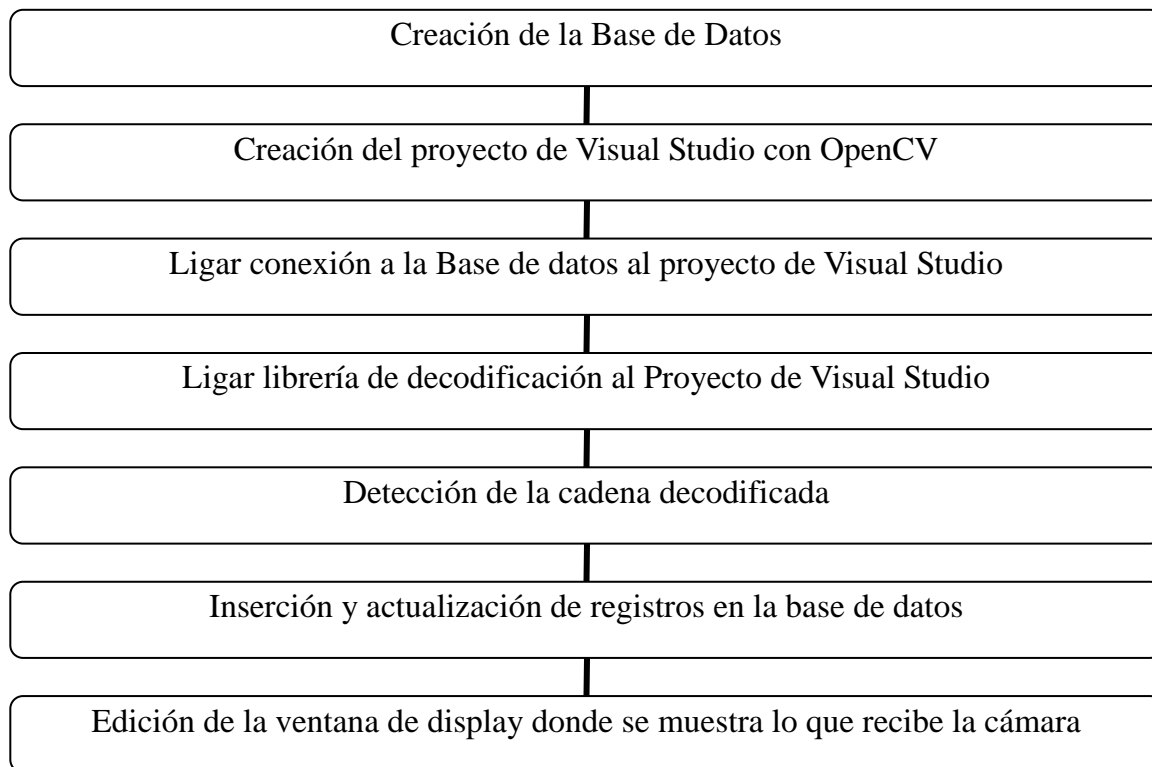
**Imagen 2-3.** Códigos QR generados.

## 2.2 Metodología de Ingeniería de Software empleada

Para el desarrollo de este proyecto se planteó el uso de Scrum, puesto que el cliente dijo que él manejaba esta metodología de desarrollo, adaptándolo en este caso al equipo de desarrollo que se conformaba de una sola persona, además de que el Product Owner y el Scrum Manager eran así mismo la misma persona.

Por la cantidad de personas que participaron en este proyecto, en base a la metodología de desarrollo, solo se manejó el Sprint Review Meeting fijada para los viernes.

En cuanto a la documentación se manejó el Product backlog como los requerimientos que dio el cliente, quedando de la siguiente manera.



### **Product Backlog.**

Los Sprint backlogs generados durante el desarrollo se pueden ver en el **apéndice A**.

El burn down chart se puede ver en el **apéndice B**.

## 2.3 Implementación de la Base de Datos

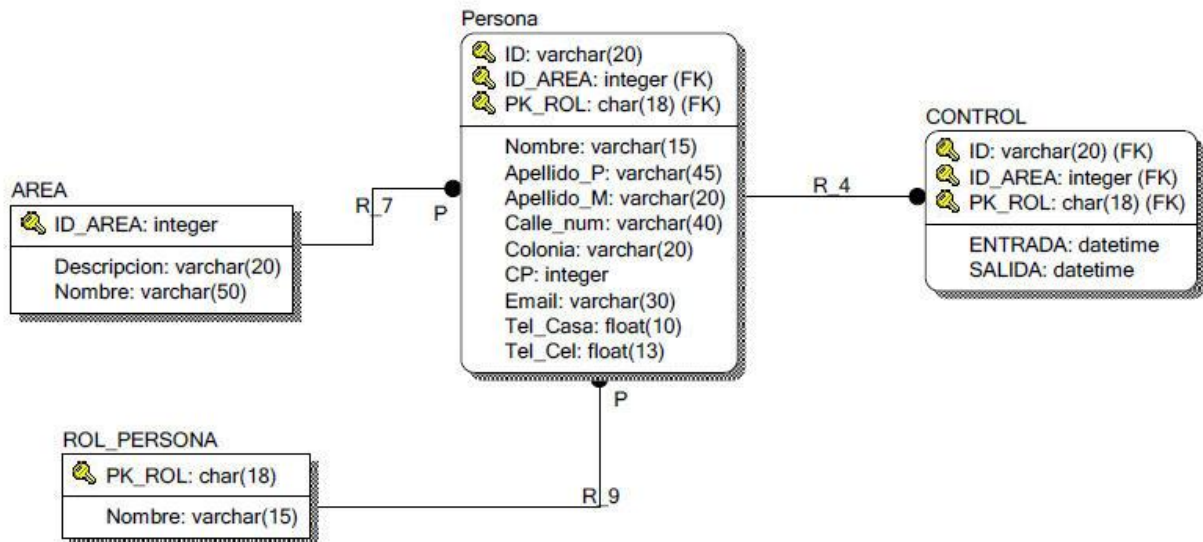
Desde el planteamiento inicial del proyecto se especificó que éste debería de ser alojado en un servidor con el cual ya se contaba, por lo tanto las decisiones referentes al alojamiento de datos y del sistema fueron tomadas en base a dichas especificaciones.

Dicho servidor era una computadora con procesador Pentium 4, sistema operativo Linux con distribución Fedora, disco duro de 80 GB, memoria ram de 1 GB.

En cuanto al software, contaba con una versión de Mysql 5.5.16, con el juego de caracteres UTF-8 Unicode (utf8). Esta versión de Mysql puede ser administrada con phpMyAdmin versiones 3.0.X en adelante, manejando en mi caso la versión 3.4.5.

El primer objeto creado fue la base de datos, la cual es una base de datos dinámica relacional bajo un sistema de gestión de base de datos o DBMS (database management system) MySQL, ésto debido a que el servidor con el que se contaba tenía instalado este tipo de manejador de base de datos.

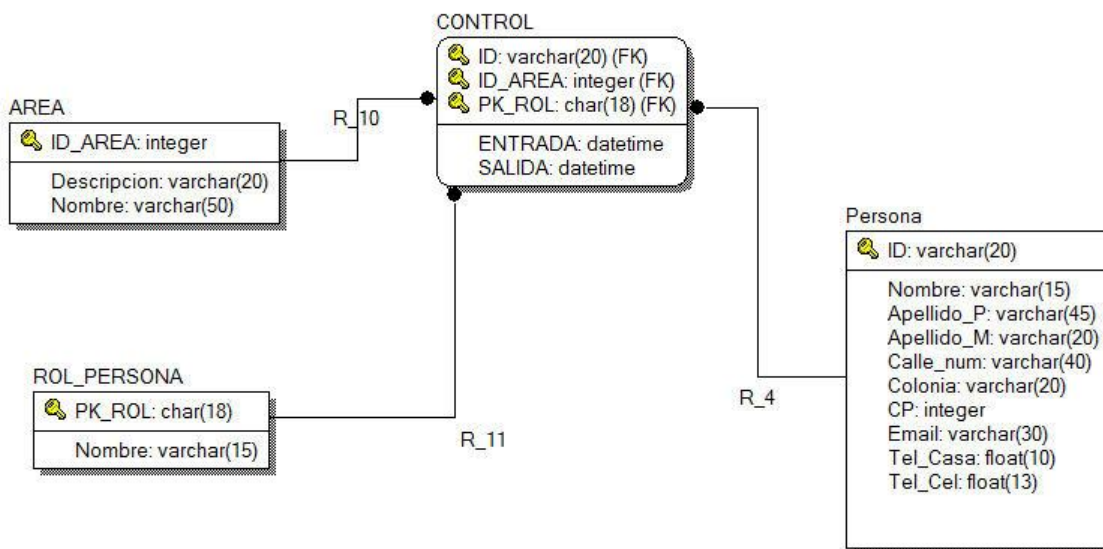
El diagrama relacional de la base de datos es el mostrado en la imagen 2-4.



**Imagen 2-4.** Diagrama relacional inicial de la BD.

Como no se tenía una base de datos en donde estuvieran los datos de todas las personas asignadas al área, se especificó que se almacenaran éstos en esta base de datos, por lo tanto, es que dichos datos están almacenados en la tabla persona; así mismo, se dijo que se incluyera una tabla de área, porque si el proyecto era exitoso, se podría extender su uso e implementarse dentro de toda la división.

En una revisión ya posterior a que el sistema estuviera funcionando, se identificó que la base de datos no estaba normalizada, si esto se hubiera hecho, la base de datos hubiera quedado como se muestra en la imagen 2-5.



**Imagen 2-5.** Diagrama de relación final de la BD.

El código para la creación de la base de datos se puede consultar en el **apéndice C**.

## 2.4 Código de la aplicación en lenguaje C++

Una vez creada la base de datos y los datos almacenados, el siguiente paso fue la implementación del código en lenguaje C++ para la aplicación.

El código tiene la siguiente estructura:

- Librerías externas que se incluyen para su uso.
  - #include <stdio.h>
- Definición de variables globales.

- using namespace std;
- SAConnection con; // connection object
- SACommand cmd; // command object
- Se crea la función main.
  - int main(int argc, char \*argv[])
- Se declara la conexión a la base de datos.
  - con.Connect (); y cmd.setConnection(&con);
- Se inicializa la cámara.
  - CvCapture \*capture=cvCaptureFromCAM(0);
- Se inicializa el decodificador.
  - QrDecoderHandle decoder=qr\_decoder\_open();
- Se crea la ventana que muestra la imagen de la cámara.
  - cvNamedWindow("src",1);
- Se crea la ventana con las instrucciones de uso.
  - cvNamedWindow("bin",1);
- Se inicializa la variable que almacenará el frame con el cual se trabajará y en la cual se aplicará el decodificador:
  - IplImage \*camera=cvQueryFrame(capture);
  - qr\_decoder\_set\_image\_buffer(decoder,src);
- Se espera a detectar una pulsación de la barra espaciadora para leer la imagen que toma la cámara.
  - if(key==0x20&&!qr\_decoder\_is\_busy(decoder))
- Se decodifica la imagen y se analiza si tiene la cadena de control.
  - char \*m;
  - m=strstr(p,DECD)
  - if(m) { }
- Si se encuentra la cadena de control, se comprueba si es un nuevo registro o es una actualización y se manda esta información a la base de datos para su almacenamiento.
  - if(control!=dia2)//modificado
  - {
  - len1=(""+dia+"",""+hora+"",""+codigo+"");
  - completa=("Insert into control(Fecha,Entrada,ID) values("+len1+"");
  - }
  - else
  - {

```

len1=(""+hora+"");
completa=("update control set Salida =" +len1+" where (ID="+codigo+" and
Fecha="+dia+"");
}

```

- Si no es la cadena de control, se dice si se pudo leer o si la imagen no pudo ser decodificada.
  - else if(len>0) // else { }
- Se dibujan los marcos de colores de acuerdo al código que se leyó.
  - cvLine(src,pt,vertexes[i],CV\_RGB(r,g,b),3);
- Se destruye la variable que almacena la imagen grabada con la cámara.
  - if(bin)

```

cvReleaseImage(&bin);
if(src)
cvReleaseImage(&src);
cvReleaseCapture(&capture);

```
- Se destruye la variable que almacena la cadena leída.
  - if(text)

```

delete text;

```
- Se reinicia la cámara a la espera de una nueva pulsación de la barra espaciadora.
  - return(0);

El código completo se puede consultar en el **apéndice D**.

## 2.5 Descripción de los pasos que sigue el programa en su ejecución

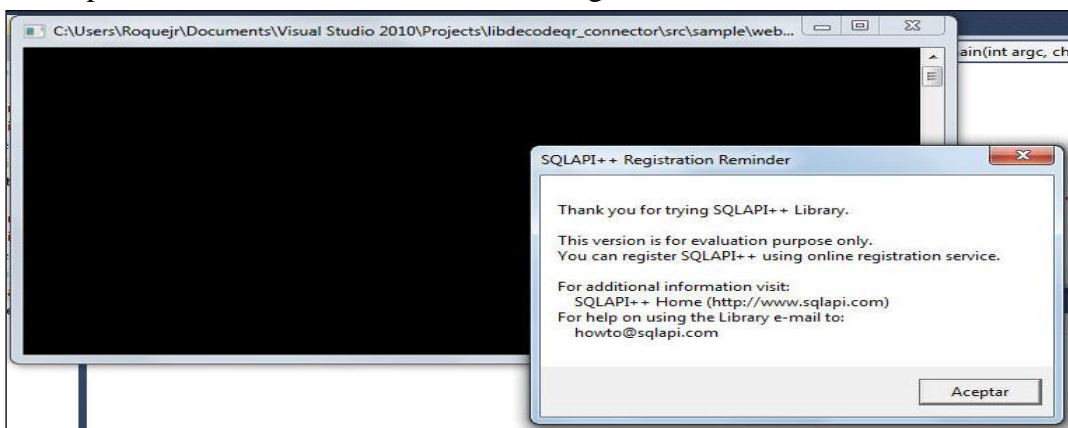
La forma en la cual trabaja la aplicación es la siguiente:

- 1 Se inicia la aplicación.
- 2 Se comprueba que haya conexión a la base de datos
  - 2.1 Si la conexión a la base de datos es fallida, se manda un mensaje de alerta y la aplicación se cierra.
- 3 Si la conexión es correcta, se muestra un mensaje con las instrucciones en una pantalla de consola.
- 4 Se inicia la cámara web en espera de que se lea algún código.
  - 4.1 Para que el código sea escaneado, debe posicionarse frente a la cámara y presionar la tecla de barra espaciadora.

- 5 Se comprueba que el código haya sido leído correctamente,
  - 5.1 Si el código no se leyó bien se pone el marco rojo diciendo que no se pudo decodificar y que se intente de nuevo.
  - 5.2 Si el código es leído, pero no se reconoce la cadena de control que indique que es un código de los usados dentro de la oficina, se muestra un marco azul y se pide que se intente de nuevo.
  - 5.3 Si el código es leído y se reconoce la cadena de control indicando que es un código de la oficina, se muestra el marco verde y se procede.
- 6 Una vez aprobado el código se almacena la cadena decodificada como una variable.
- 7 Se obtiene del sistema la fecha y hora y se almacenan en una variable.
- 8 Se ejecuta una sentencia sql usando como parámetros la variable del código decodificado y de la variable con la fecha y hora.
- 9 Se comprueba si existe dentro de la tabla de control un registro que tenga la fecha y la cadena decodificada
  - 9.1 Si no hay registro, entonces se crea y se pone la hora almacenada en el campo de hora de entrada.
  - 9.2 Si ya existe registro, entonces el dato de hora se almacena en el campo de hora de salida de ese registro existente.
- 10 Se recibe desde la base de datos una bandera de que se hizo un movimiento en ésta.
- 11 Se pone un mensaje en la pantalla de que ha sido modificado el registro y la aplicación se regresa al punto 1, a la espera de otro escaneo.

Las pantallas del programa en funcionamiento son las siguientes

En caso de ser exitosa la conexión con la base de datos, se muestra el mensaje de uso de la librería usada para conectar con la base de datos. Imagen 2-6.



**Imagen 2.6** Pantalla de conexión a la BD.

Imagen 2-7. Una vez que se comprobó la conexión a la base de datos en ésta se despliegan las instrucciones de cómo funciona el programa y el procedimiento a seguir.

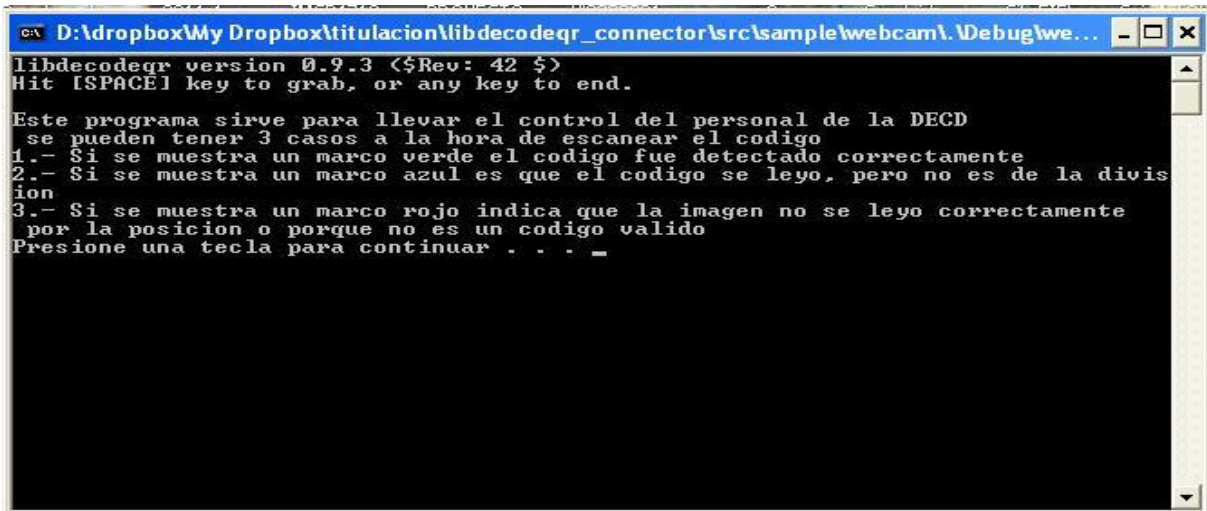


Imagen 2-7. Pantalla de las instrucciones del programa.

Imagen 2-8. Pantalla que se despliega en caso de que no sea detectado correctamente el código, como se puede ver, indica los datos de la imagen que se trató de decodificar y el marco del código se pone en color rojo.

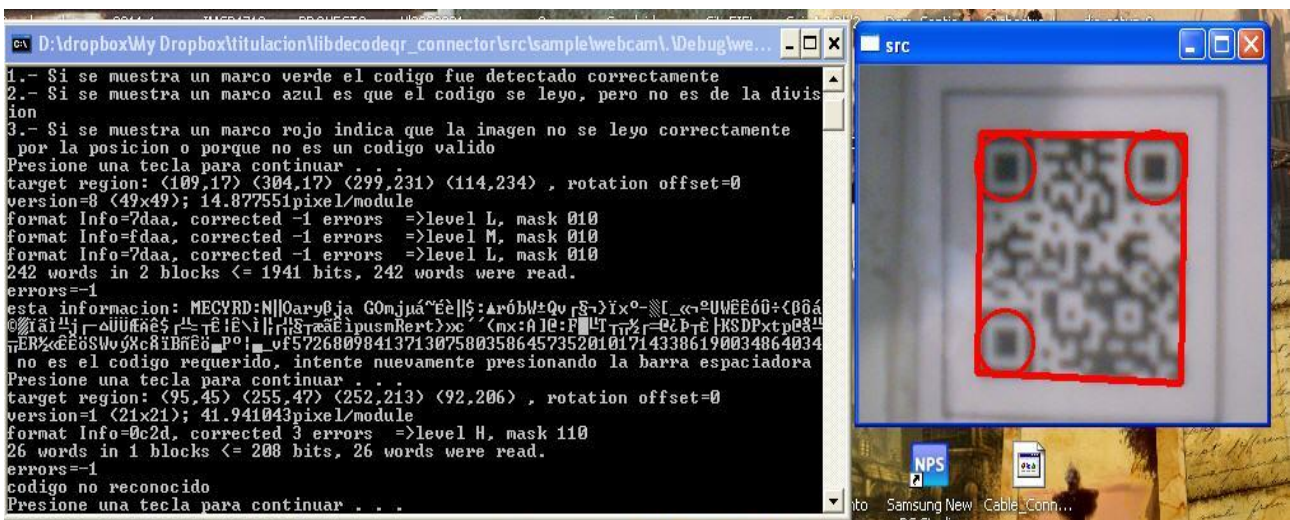
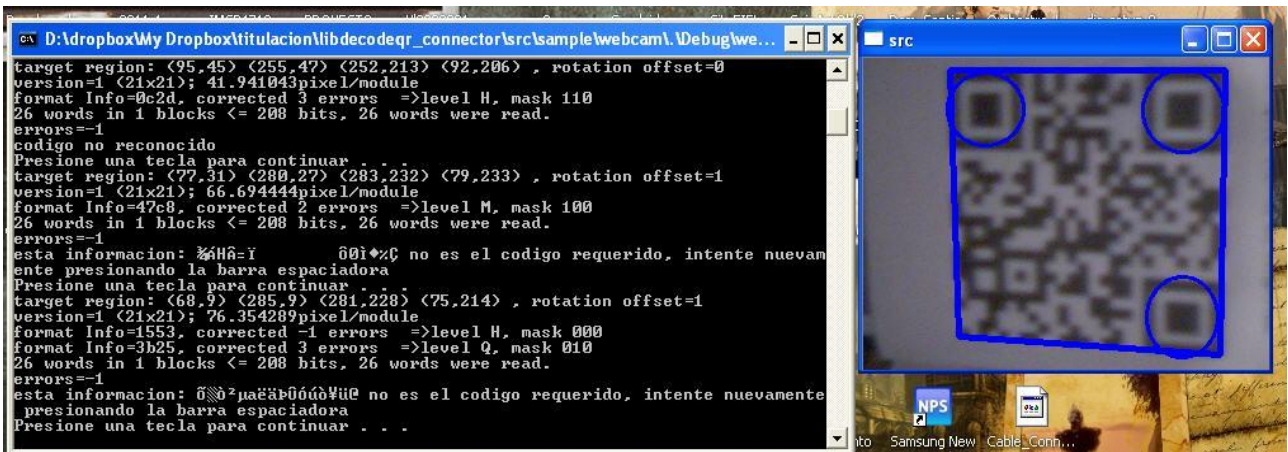


Imagen 2-8. Pantalla de detección de código incorrecto.

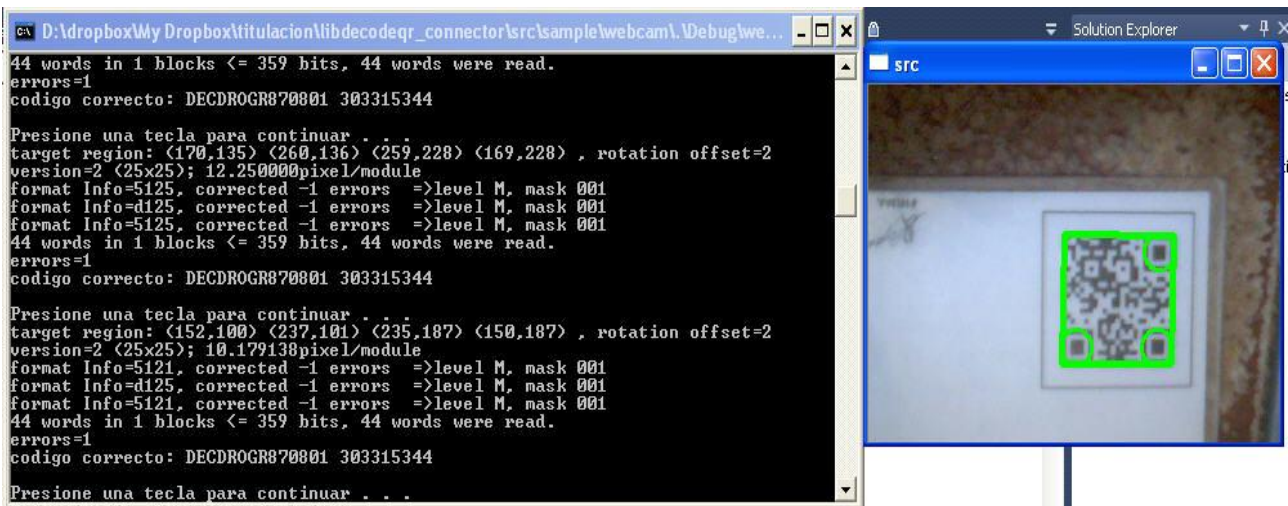
Imagen 2-9. El segundo caso, cuando se lee correctamente el código, pero la información descryptada no es la cadena de control de la oficina, el marco del código se remarca de color azul.





**Imagen 2-9.** Pantalla de detección de código que no es el de la DECDFI.

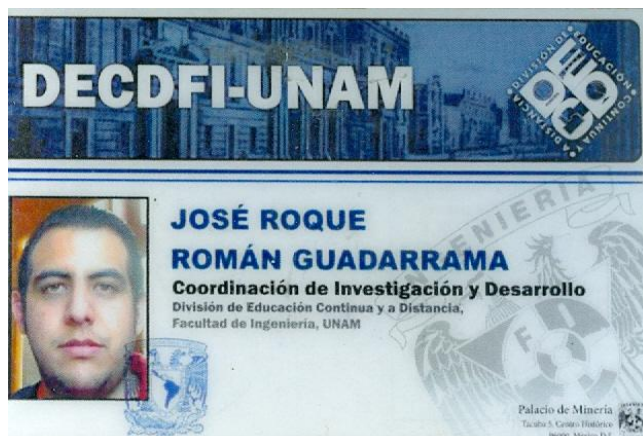
Imagen 2-10. El tercer caso, en el cual se detecta correctamente el código y además la información descryptada es la cadena de control, se muestra la leyenda código correcto seguida del código leído y se marca en color verde el marco del código.



**Imagen 2-10.** Pantalla de detección de código que es de la DECDFI.

## 2.6 Diseño de las credenciales

El diseño frontal de las credenciales se muestra en la imagen 2-11. Y el diseño trasero es el mostrado en la imagen 2-12.



**Imagen 2-11.** Diseño frontal de la credencial.



**Imagen 2-12.** Diseño trasero de la credencial.

En este caso, los códigos QR generados y utilizados tienen contenida la cadena de control en texto plano, es decir, que no se usó ninguna codificación extra como medida de seguridad. Por lo tanto, la única medida que se usa para corroborar que el usuario es el propietario de la credencial es la fotografía.

### 3 Resultados obtenidos del proyecto

El periodo de prueba del sistema duró 5 meses, estos son los resultados que se obtuvieron:

- El sistema fue implementado dentro del área de I+D de la División, en esta área laborábamos alrededor de 21 personas.
- El sistema fue usado por 4 becarios, 5 trabajadores por honorarios, 4 tesistas y 8 prestadores de Servicio.
- En promedio se generaron 28 registros por persona durante el periodo que el sistema estuvo activo.
- El mínimo de registros generados por una persona fue de 2.
- El máximo de registros generados por una persona fue de 64.
- 9 personas tuvieron más de 28 registros y generaron más del 77% de éstos.
- El promedio de horas laboradas de acuerdo a los registros obtenidos por el programa son:
  - Tesistas: 6 horas. Los 4 tesistas excedieron las horas establecidas.
  - Prestadores de Servicio Social: 3 horas 50 minutos. 2 personas cubrieron las horas establecidas
  - Becarios: 4 horas 5 minutos. 2 personas cubrieron las horas establecidas
  - Honorarios: 5 horas 50 minutos.
- Se esperaban un total de 1344 registros y solo se obtuvieron 581, con una eficiencia del 43 %; esto debido a que el uso del sistema lo dejó como opcional el jefe de la división y por ello de acuerdo a los promedios de horarios y de horas registradas por el sistema los becarios y los prestadores de Servicio Social fueron los usuarios que más usaron el sistema, mientras que los trabajadores de honorarios no usaban el sistema continuamente.

## 4 CONCLUSIONES

El proyecto de acceso de entrada y salida con el uso de código QR arrojó resultados positivos.

Este proyecto sentó bases para proyectos futuros; un ejemplo de esto es la posibilidad de incluir los códigos QR como una forma de identificación en los diplomas y/o reconocimientos de los cursos que imparte la DECD.

Si los registros obtenidos por el sistema durante el periodo de prueba hubieran sido utilizados por el departamento de Contabilidad para los pagos de las becas, los honorarios y el requisito de horas para el Servicio Social, se hubieran dado los siguientes casos:

- A todos los becarios se les pagaba casi la totalidad de la beca
- Los prestadores de Servicio Social en lugar de cubrir el total de sus horas en 6 meses lo harían en 12 meses
- Los trabajadores de honorarios hubieran recibido casi todos la mitad de su paga, de acuerdo a las horas que laboraron.

Fue una gran experiencia laborar en el área de I+D y sobre todo el que hubiera escogido la rama de Visión Computacional y Realidad Aumentada, porque esto significó una gran cantidad de investigación, desarrollo y aplicación de conocimientos y tecnologías nuevos para mí o que solo había usado teóricamente, lo cual no siempre se puede obtener en un trabajo.

El aprender a manejar la librería SQL API fue muy bueno, puesto que yo no tenía conocimientos de alguna librería de conexión de base de datos con C++ y esto es uno de los grandes requerimientos que se buscan en casi todos los proyectos en el ámbito laboral, es muy raro el caso en el que no se necesite una conexión a una base de datos.

El uso de la librería OpenCV fue una gran adición a los conocimientos básicos que tenía de ésta, permitiéndome de esta forma tener un mayor alcance de desarrollo al que tenía previamente.

Así mismo el uso de la librería libdecodeqr me sirvió para conocer un poco el proceso de como se pueden decodificar estos códigos y que uso se les puede dar.

Como experiencia obtenida puedo decir que aprendí a usar nuevas librerías, más profundización en programación con el lenguaje C++, aplicación de diversos conocimientos adquiridos en la carrera de Ingeniería de Computación, tanto de las materias de Ciencias Básicas, las de la carrera y las del módulo de Computación Gráfica. Entre estos conocimientos están los adquiridos en Computación para Ingenieros, Ingeniería de Software, Bases de Datos, Procesamiento Digital de Imágenes, Computación Gráfica Avanzada, Algoritmos y Estructuras de Datos por mencionar los más importantes.

## 4.1 Proyección a Futuro

Cuando se habló del desarrollo de este proyecto, se comentó la posibilidad de que una vez que se usara y se comprobara que su desempeño era bueno, se extendiera el uso a las otras áreas de la División, así como la adaptación de este sistema para el control de los diplomas y constancias que la División entrega a los alumnos que toman los cursos que imparte.

En caso de implementarse esta extensión del sistema se debe de considerar un método de seguridad extra, para que se pueda comprobar que los diplomas son auténticos.

Una forma de implementar esta medida de seguridad es que la cadena de control se codifique y el resultado que se obtenga sea el que se almacene en el código QR, de esta forma es más difícil que se pueda presentar un código apócrifo.

## 5 APÉNDICES

### 5.1 APÉNDICE A

#### Sprint backlogs generados durante el desarrollo de la Aplicación.

##### Sprintlog 1:

Requisito	Tarea	Estatus
Creación de la Base de Datos	Crear diagrama E-R de la base de datos	Completada
Creación de la Base de Datos	Crear tablas de la base de datos	Completada
Creación de la Base de Datos	Normalización de la base de datos	Completada
Creación de la Base de Datos	Vaciado de datos en las tablas de la base de datos	Completada
Creación del proyecto de Visual Studio con OpenCV	Añadir librerías de OpenCV a la solución y dependencias, comprobar su funcionamiento correcto	Completada
Ligar conexión a la base de datos al proyecto de Visual Studio	Añadir librería de SQL API a la solución y dependencias, comprobar su funcionamiento	Completada
Ligar conexión a la base de datos al proyecto de Visual Studio	Unificar con el código principal del programa para su uso	No iniciada
Ligar librería de decodificación al proyecto de Visual Studio	Añadir librería libdecodeqr y dependencias	Completada
Ligar librería de decodificación al proyecto de Visual Studio	Unificar con el código principal del programa para su uso	En progreso
Detección de la cadena codificada	Programar la detección de la cadena de codificación	No iniciada

## Sprintlog 2:

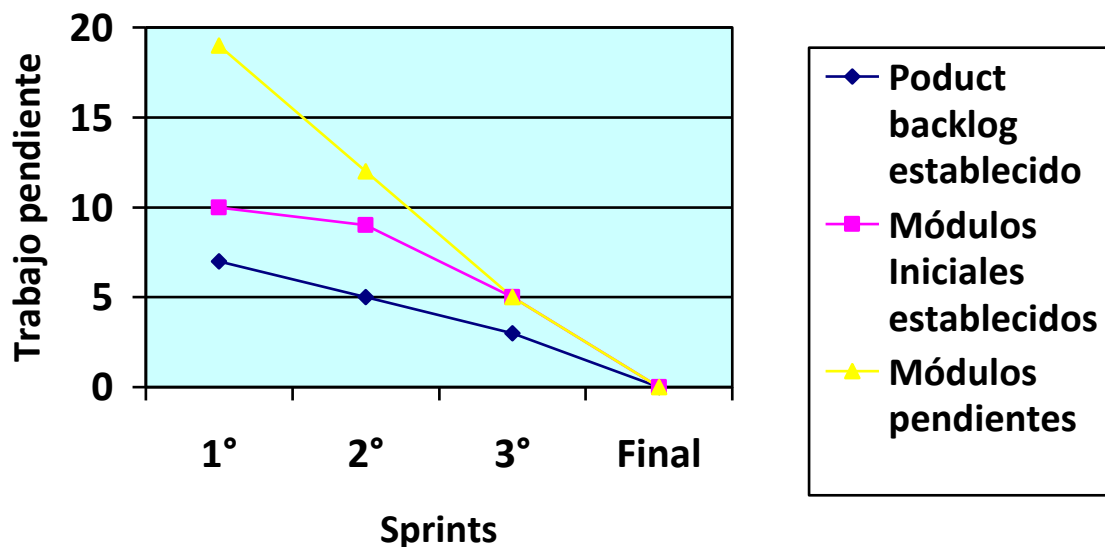
Requisito	Tarea	Estatus
Ligar librería de decodificación al proyecto de Visual Studio	Unificar con el código principal del programa para su uso	Completada
Detección de la cadena codificada	Programar la detección de la cadena de codificación	Completada
Ligar conexión a la base de datos al proyecto de Visual Studio	Unificar con el código principal del programa para su uso	Completada
Inserción y actualización de registros en la base de datos	Insertar un nuevo registro al leer la cadena de control	Completada
Inserción y actualización de registros en la base de datos	Actualizar un registro existente al leer la cadena de control	En progreso
Edición de la ventana de display donde se muestra lo que recibe la cámara	Mandar mensaje de que el código es uno de los de control o no se leyó	Completada
Edición de la ventana de display donde se muestra lo que recibe la cámara	Mandar mensaje de que el código es correcto, es diferente o no se leyó correctamente	Completada
Edición de la ventana de display donde se muestra lo que recibe la cámara	Mostrar mensaje de que se insertó el registro en la base de datos	Completada
Edición de la ventana de display donde se muestra lo que recibe la cámara	Mostrar los marcos en verde, azul o rojo, dependiendo el caso de la lectura del código	No iniciada

### Sprintlog 3:

Requisito	Tarea	Estatus
Inserción y actualización de registros en la base de datos	Actualizar un registro existente al leer la cadena de control	Completada
Actualizar un registro existente al leer la cadena de control	Checar si ya hay registro del día en la base de datos	Completada
Actualizar un registro existente al leer la cadena de control	Comprobar que el mes sea menor a 10 y agregar un 0 antes para evitar errores del mes	Completada
Edición de la ventana de display donde se muestra lo que recibe la cámara	Mostrar los marcos en verde, azul o rojo, dependiendo el caso de la lectura del código	Completada
Probar la aplicación en otra computadora	Copiar dlls y dependencias necesarias para la ejecución de la aplicación en otra computadora	Completada

## 5.2 APÉNDICE B

### Burndown chart





## 5.3 APÉNDICE C

### Código de creación de la base de datos

```
-- phpMyAdmin SQL Dump
-- version 3.2.2.1deb1
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 19-05-2011 a las 05:27:39
-- Versión del servidor: 5.1.37
-- Versión de PHP: 5.2.10-2ubuntu6.4

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT=0;
START TRANSACTION;

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `pruebas`
--
-----
--
-- Estructura de tabla para la tabla `area`
--
-- Creación: 18-10-2010 a las 09:13:49
--

DROP TABLE IF EXISTS `area`;
CREATE TABLE IF NOT EXISTS `area` (
  `ID_AREA` int(11) NOT NULL,
  `Nombre` varchar(50) NOT NULL,
  `Descripcion` varchar(20) NOT NULL,
  PRIMARY KEY (`ID_AREA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Volcar la base de datos para la tabla `area`
--
INSERT INTO `area` (`ID_AREA`, `Nombre`, `Descripcion`) VALUES
(1, 'Coordinacion de I+D', 'area de desarrollo'),
(2, 'Jefatura', 'oficina de Ernesto Riestra'),
(3, 'Coordinacion de Procesos Educativos', 'Enfocado a educación'),
(4, 'Coordinacion de Proyectos Estrategicos', 'Estrategias nuevas'),
(5, 'Coordinacion de Sistemas', 'Funcionamiento de re'),
(6, 'Coordinacion del Palacio de Minería', 'General'),
(7, 'Secretaria Academica', 'parte academica'),
(8, 'Unidad Administrativa', 'parte administrativa');
```

-----

```
--
-- Estructura de tabla para la tabla `control`
--
-- Creación: 27-01-2011 a las 06:03:11
-- Última actualización: 19-05-2011 a las 05:20:46
-- Última revisión: 02-05-2011 a las 01:39:21
--
```

```
DROP TABLE IF EXISTS `control`;
CREATE TABLE IF NOT EXISTS `control` (
  `Fecha` date NOT NULL,
  `Entrada` time DEFAULT NULL,
  `Salida` time DEFAULT NULL,
  `ID` varchar(20) NOT NULL,
  KEY `ID` (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
-- RELACIONES PARA LA TABLA `control`:
-- `ID`
-- `persona` -> `ID`
--
--
-- Volcar la base de datos para la tabla `control`
```

```

--
-
-- Estructura de tabla para la tabla `persona`
--
-- Creación: 19-10-2010 a las 09:42:44
--

DROP TABLE IF EXISTS `persona`;
CREATE TABLE IF NOT EXISTS `persona` (
  `ID` varchar(20) NOT NULL,
  `Nombre` varchar(15) NOT NULL,
  `Apellido_P` varchar(20) NOT NULL,
  `Apellido_M` varchar(20) NOT NULL,
  `Calle_NUM` varchar(40) NOT NULL,
  `CP` int(11) NOT NULL,
  `Colonia` varchar(30) NOT NULL,
  `Tel_Casa` int(11) DEFAULT NULL,
  `Tel_Cel` bigint(20) DEFAULT NULL,
  `Hora_ingreso` time NOT NULL,
  `Hora_salida` time NOT NULL,
  `ID_AREA` int(11) NOT NULL,
  `PK_ROL` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `P_FK_A` (`ID_AREA`),
  KEY `P_FK_R` (`PK_ROL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- RELACIONES PARA LA TABLA `persona`:
--  `ID_AREA`
--  `area` -> `ID_AREA`
--  `PK_ROL`
--  `rol_persona` -> `PK_ROL`
--
--
-- Volcar la base de datos para la tabla `persona`
--

INSERT INTO `persona` (`ID`, `Nombre`, `Apellido_P`, `Apellido_M`, `Calle_NUM`, `CP`, `Colonia`,

```

```
`Tel_Casa`, `Tel_Cel`, `Hora_ingreso`, `Hora_salida`, `ID_AREA`, `PK_ROL`) VALUES  
  
(`DECDROGR870801`, `Jose Roque`, `Roman`, `Guadarrama`, `Hacienda de Pasteje N°31-302`, 14310, `Floresta  
Coyoacan`, 56846502, 5529079423, `09:00:00`, `13:00:00`, 1, 1;
```

-----

```
--  
-- Estructura de tabla para la tabla `rol_persona`
```

```
--  
-- Creación: 19-10-2010 a las 05:06:07
```

```
--  
  
DROP TABLE IF EXISTS `rol_persona`;  
CREATE TABLE IF NOT EXISTS `rol_persona` (  
  `PK_ROL` int(11) NOT NULL,  
  `Nombre` varchar(25) NOT NULL,  
  PRIMARY KEY (`PK_ROL`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Volcar la base de datos para la tabla `rol_persona`
```

```
--  
  
INSERT INTO `rol_persona` (`PK_ROL`, `Nombre`) VALUES  
(1, `becario`),  
(2, `honorarios`),  
(3, `Tesista`),  
(4, `Base`),  
(5, `Servicio Social`),  
(6, `Practicas Profesionales`);
```

```
--  
-- Filtros para las tablas descargadas (dump)
```

```
--  
  
-- Filtros para la tabla `persona`
```

```
--  
ALTER TABLE `persona`
```

```

ADD CONSTRAINT `P_FK_A` FOREIGN KEY (`ID_AREA`) REFERENCES `area` (`ID_AREA`) ON DELETE
CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `P_FK_R` FOREIGN KEY (`PK_ROL`) REFERENCES `rol_persona` (`PK_ROL`) ON
DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

```

## 5.4 APÉNDICE D

### Código de la aplicación

```

////////////////////////////////////
//
// webcam.cpp --a part of libdecodeqr
//
// Copyright(C) 2007 NISHI Takao <zophos@koka-in.org>
//           JMA (Japan Medical Association)
//           NaCl (Network Applied Communication Laboratory Ltd.)
//
// This is free software with ABSOLUTELY NO WARRANTY.
// You can redistribute and/or modify it under the terms of LGPL.
//
#include <cv.h>
#include <cxcore.h>
#include <stdio.h>
#include <string.h>
#include <highgui.h>
#include <windows.h>
#include "../libdecodeqr/decodeqr.h"
#include <SQLAPI.h>
#include <WinSock.h>
#include <myAPI.h>
#include <stdlib.h>
#include <iostream>
#include <string>
#include <ctime>
using namespace std;
SAConnection con; // connection object
SACommand cmd; // command object

int usage(char *program_name);

int main(int argc, char *argv[])
{
    con.Connect(

```

```

        "localhost@pruebas", // database name
        "ar_cv", // user name
        "admin_cv", // password
        SA_MySQL_Client); //client name

    /*con.Connect(
        "192.168.2.133@pruebas", // database name
        "ar_cv", // user name
        "admin_cv", // password
        SA_MySQL_Client); //client name
    */
cmd.setConnection(&con);
    int show_bin_image=0;

if(argc>1){
    if(strcmp(argv[1],"-d"))
        return(usage(argv[0]));
    else
        show_bin_image=1;
}

// start camera
//
//webcam externa CvCapture *capture=cvCaptureFromCAM(0);
    CvCapture *capture=cvCaptureFromCAM(1); //webcam de la laptop
if(!capture)
    return(-1);

//
// initialize qr decoder
//
QrDecoderHandle decoder=qr_decoder_open();
printf("libdecodeqr version %s\n",qr_decoder_version());

cvNamedWindow("src",1);

if(show_bin_image)
    cvNamedWindow("bin",1);

puts("Hit [SPACE] key to grab, or any key to end.");
puts("");
    printf("Este programa sirve para llevar el control del personal de la DECD\n se pueden
tener 3 casos a la hora de escanear el codigo\n");
    printf("1.- Si se muestra un marco verde el codigo fue detectado correctamente\n");
    printf("2.- Si se muestra un marco azul es que el codigo se leyo, pero no es de la
division\n");
    printf("3.- Si se muestra un marco rojo indica que la imagen no se leyo correctamente\n por
la posicion o porque no es un codigo valido\n");
    system("PAUSE");

// allocate grabed buffer to decoder

```

```

int key=-1;

IplImage *camera=cvQueryFrame(capture);
IplImage *src=NULL,*bin=NULL;
if(camera){
    src=cvCloneImage(camera);
    qr_decoder_set_image_buffer(decoder,src);
}
else
    key=1;

unsigned char *text=NULL;
int text_size=0;
while(key<=0){
    cvShowImage("src",camera);
    key=cvWaitKey(150);

    //
    // when [SPACE] key pressed, do decode.
    //
    if(key==0x20&&!qr_decoder_is_busy(decoder))
        {
            key=-1;

            //
            // if left-bottom origin (MS-Windows style) format,
            // it must be converted to left-top origin.
            //
            if(camera->origin)
                cvConvertImage(camera,src,CV_CVTIMG_FLIP);
            else
                cvCopy(camera,src);

            //
            // While decoding is a failure, decrease the
            // adaptive_th_size parameter.
            // Note that the adaptive_th_size must be odd.
            //
            short sz,stat;
            for(sz=25,stat=0;
                (sz>=3)&&((stat&QR_IMAGEREADER_DECODED)==0);
                sz-=2)
                stat=qr_decoder_decode(decoder,sz);

            //
            // for debug, show binarized image.
            //
            if(bin)
                cvReleaseImage(&bin);
            if(show_bin_image){
                bin=cvCloneImage(qr_decoder_get_binarized_image_buffer(decoder));
                cvShowImage("bin",bin);
            }
        }
}

```

```

}
// printf("adaptive_th_size=%d, status=%04x\n",sz,stat);

//
// on succeed decoding, print decoded text.
//
QRCodeHeader header;
if(qr_decoder_get_header(decoder,&header)){
    if(text_size<header.byte_size+1){
        if(text)
            delete text;

        text_size=header.byte_size+1;
        text=new unsigned char[text_size];
    }
    qr_decoder_get_body(decoder,text,text_size);
    printf("%s\n\n",text);
    //parser

    char *p=(char *)text;
    int len;
    len=strlen(p);
    char DECD[]="DECD";
    char *m;
    m=strstr(p,DECD);
    int color=0;

    if(m)
    {

        string len1;
        string entrada;
        entrada=p;
        string codigo;
        codigo=entrada.substr(0,14);
        string completa;
        char date[9];
        char time[9];
        string dia;
        string fecha;
        string hora;
        dia=(_strdate(date));
        string c;
        c=dia.substr(6,8);
        for(int a=1;a<4;a++)
            dia.pop_back();
        dia=c+"/"+dia;
        //cambiar formato de dia
        hora=(_strtime(time));
        string control;
        control="select Fecha from control
where(ID='"+codigo+"'and Fecha='"+dia+"')";

```



```

const char *comando;
//const char *comando2;
comando=control.c_str();
SAString comando_final(comando);
cmd.setCommandText(comando_final);
cmd.Execute();
/*
if(cmd.isResultSet())
{
    len1((""+hora+ "");
    completa("update control set Salida =" +len1+"
where (ID="" +codigo+ "" and Fecha="" +dia+ "")");
}
else
{
    len1((""+dia+ "", ""+hora+ "", ""+codigo+ "");
    completa("Insert into
control(Fecha,Entrada,ID) values("+len1+"");
}

*/

SADateTime fecha_check;
cmd.FetchNext();
// fecha_check=(const char*)cmd.Field("Fecha").asString();
fecha_check=cmd.Field("Fecha").asDateTime();
int año=fecha_check.GetYear();
char year[5]="";
char month[3]="";
char daily[3]="";
sprintf(year,"%d",año);
string y=year;
int mes=fecha_check.GetMonth();
//esto me causo muchos problemas
if (mes<10)
sprintf(month,"0%d",mes);
else
sprintf(month,"%d",mes);
string m=month;
int day=fecha_check.GetDay();
sprintf(daily,"%d",day);
string d=daily;
string control2;
control2=y+"/"+m+"/"+d;
string dia2="20"+dia;
//string fecha_control=fecha_check.operator SAString();
if(control2!=dia2)//modificado
{
    len1((""+dia+ "", ""+hora+ "", ""+codigo+ "");
    completa("Insert into
control(Fecha,Entrada,ID) values("+len1+"");
}

```

```

else
{
    len1=(""+hora+""");
    completa=("update control set Salida =" + len1 + "
where (ID="" + codigo + "" and Fecha="" + dia + """);
}

comando=completa.c_str();
SAString comando_final2(comando);
cmd.setCommandText(comando_final2);
cmd.Execute();
// commit changes on success
con.Commit();
color=1;
printf("codigo correcto: %sz\n\n",text);
//printf("dia es: %s",dia);

}
else if(len>0)
{

    printf("esta informacion: %s no es el codigo requerido, intente nuevamente
presionando la barra espaciadora\n",p);
    color=2;
}
else
{
    printf("codigo no reconocido \n");
}
//
//
// draw found code region with colored line
//
int r=0;
int g=0;
int b=0;
if(color==1)
g=255;
else if(color==2)
b=255;
else
r=255;
CvPoint *vertexes=qr_decoder_get_coderegion_vertexes(decoder);
CvPoint pt=vertexes[3];
int i;
for(i=0;i<4;i++){
    cvLine(src,pt,vertexes[i],CV_RGB(r,g,b),3);
    pt=vertexes[i];
}
//
// draw found finder patterns with colored ellipse
//

```

```

    CvBox2D *boxes=qr_decoder_get_finderpattern_boxes(decoder);
    for(i=0;i<3;i++){
        CvSize sz=cvSize((int)boxes[i].size.width/2,
            (int)boxes[i].size.height/2);
        cvEllipse(src,
            cvPointFrom32f(boxes[i].center),
            sz,
            boxes[i].angle,
            0,360,
            CV_RGB(r,g,b),2);
    }

    if(src->origin)
        cvConvertImage(src,src,CV_CVTIMG_FLIP);

    cvShowImage("src",src);

        key=cvWaitKey(1500);
        system("PAUSE");
    }

}

camera=cvQueryFrame(capture);
if(!camera)
    break;
}

if(text)
    delete text;

qr_decoder_close(decoder);
if(bin)
    cvReleaseImage(&bin);
if(src)
    cvReleaseImage(&src);

cvReleaseCapture(&capture);

return(0);
}

int usage(char *program_name)
{
    fprintf(stderr,"usage: %s [-d|-h]\n",program_name);
    fprintf(stderr,"-d\tturn on debug mode.\n");
    fprintf(stderr,"-h\tshow thismessage and quit.\n\n");

    return(-1);
}

```

## 6 GLOSARIO

1. **Visión Computacional:** Rama de la Inteligencia Artificial que se incluye métodos para analizar y comprender imágenes y trasladarlas a datos que la computadora puede procesar.
2. **Realidad Aumentada:** Uso de dispositivos para agregar elementos virtuales que se pueden contemplar visualmente en la realidad, para producir una realidad mixta en tiempo real.
3. **Código QR:** Código Quick response, imagen que contiene datos en un arreglo matricial de puntos, también denominado código de barras bidimensional.
4. **API:** Interfaz de Programación de Aplicaciones, conjunto de funciones y procedimientos que contiene una biblioteca, para poder ser usado por otro software.
5. **C++:** Lenguaje de programación.
6. **OpenCV:** Librería que permite manejar operaciones de Visión Computacional en el software desarrollado, es compatible con los lenguajes de programación C, C++, Python, Ruby, Matlab.
7. **GUI:** Interfaz gráfica del usuario, componente de los lenguajes de programación y bibliotecas, necesario para crear un elemento gráfico que sirva de interfaz para el usuario a con el software.
8. **Base de datos:** Conjunto de datos almacenados de forma sistemática para su posterior consulta.
9. **DBMS:** Sistema manejador de base de datos, interfaz para que el usuario pueda acceder a una base de datos y manipular o consultar los datos que esta almacena.
10. **MySQL:** En un sistema manejador de base de datos, muy usado para aplicaciones web.

## 7 REFERENCIAS

---

<sup>1</sup> DECDFI [en línea] [consulta: 7 Diciembre 2011 ] disponible en:  
<http://www.mineria.unam.mx/historia.php>

<sup>2</sup> Welcome to opencv documentation! [en línea] [consulta: 17 Diciembre de 2011 ]  
disponible en: <http://docs.opencv.org/>

<sup>3</sup> Bradski, Gary. *Learning OpenCV Computer Vision with the OpenCV Library*, Ed.  
O' Reilly Media 2008, pps 578

<sup>4</sup> ISO/IEC18004 1st edition 2000-06-15 [en línea] [consulta: 18 Diciembre de 2011]  
disponible en [http://raidenii.net/files/datasheets/misc/qr\\_code.pdf](http://raidenii.net/files/datasheets/misc/qr_code.pdf)

<sup>5</sup> SQLAPI++ Library [en línea] [consulta: 2 Diciembre 2011] disponible en:  
<http://www.sqlapi.com/>

<sup>6</sup> libdecodeqr [en línea] [consulta: 15 Diciembre 2011] disponible en <http://trac.koka-in.org/libdecodeqr>

<sup>7</sup> Principios del Manifiesto Ágil [en línea] [consulta: 15 Enero 2013] disponible en:  
<http://www.agilemanifesto.org/iso/es/principles.html>

<sup>8</sup> Levey, Jhonatan. Not All QR Codes are Created Equal [en línea] [consulta: 15  
Marzo 2012] disponible en: <http://www.fathomdelivers.com/blog/index.php/not-all-qr-codes-are-created-equal/>