

**Sistema móvil de detección de plazas de estacionamiento en la plataforma del proyecto tequila pueblo mágico inteligente**

**Prototipo de detección de plazas**

**REPORTE DE PRACTICA**

**HARDWARE, REDES Y DISPOSITIVOS IOT**

**PRESENTA**

**Ing. José Luis Cobián Hermosillo**

**Profesor**

**Dr. Rubén Estrada Marmolejo**

**Junio 2019**

**URL DEL CODIGO: <https://github.com/cobymotion/PrototipoIOT>**

**PRESENTACIÓN**

Tequila, pueblo mágico desde 2003, con una población de alrededor de 41,000 habitantes, quienes tienen como medio de sustento alguna actividad relacionada con la bebida tradicional, siendo esta de gran atractivo tanto para las personas cercanas como miles de turistas que visitan a tequila, sin embargo, también existen diferentes atractivos tanto religiosos como históricos que hacen al lugar, uno de los más concurridos en el estado y país. Esto hace que el turismo sea una de las principales actividades de sustento de muchos de los habitantes del pueblo mágico, esto conlleva grandes posibilidades para el ciudadano para obtener una fuente ingreso.

Tomando en cuenta lo anterior Tequila al ser una localidad pequeña y con una infraestructura urbana modesta por su distinción como pueblo mágico tiene problemas de afluencia vehicular por la cantidad de turistas que visitan al lugar. En este aspecto CODIT en conjunto con el gobierno y empresarios están impulsando proyectos de tecnológica que permitan mejorar varios aspectos que mejoren la calidad de vida de los habitantes entre ellos el de movilidad que es eje fundamental de la investigación, en este aspecto, las aplicaciones tecnológicas permiten tomar mejores decisiones aplicados a las grandes urbes. La investigación aplicada pretende ayudar a disminuir la afluencia vehicular por medio del uso de tecnología.

La plataforma, así como el dispositivo para estacionamientos tiene como objetivo solucionar aspectos de movilidad y así disminuir factores ambientales atribuibles al trafico dentro de la ciudad. La aplicación permitirá que los conductores conocer lugares donde pueden dejar su automóvil de forma de segura y cercano a su sitio a donde se dirigen, dicha aplicación se utilizara como plataforma para el dispositivo de detección de plazas, el cual permitirá avisar junto con su algoritmo de inteligencia las mejores posibilidades que tenga el visitante para poder encontrar un buen lugar donde se encontrara su vehículo.

Esta investigación se centra en el diseño y el algoritmo adaptable a la plataforma para poder mejorar aspectos que tienen que ver con la movilidad, implementándose a la par con la plataforma de estacionamientos.

Estos elementos permitirán tomar los datos referentes a las ventajas del proyecto y a su utilización, las ventajas y el costo beneficio que pueden tener al aplicar este tipo de dispositivos en un entorno de ciudad inteligente como lo esta tequila.

**RESUMEN**

En esta propuesta de proyecto se presenta los pormenores de un prototipo que permita detectar vehículos en lugares de estacionamiento para proporcionar información en tiempo real, permitiéndoles encontrar un lugar rápidamente para su vehículo con el objetivo de evitar la carga vehicular en lugares complicados como el centro o primer cuadro de la ciudad.

El dispositivo es de bajo costo y consumo, permite adaptarse a una red de información que analiza los datos, los cuales generan sugerencias a los usuarios de la aplicación móvil de estacionamientos en base a sus requerimientos o búsquedas, provocando así que se encuentre rápidamente un lugar y despejar las vías de tránsito lo más pronto posible.

El presente prototipo abarca solo una parte interna a un estacionamiento de los que el proyecto permitira sensar el cual sera equipado en cada uno de los cajones para que puedan estar mandando información en tiempo real.

**Palabras Clave: IOT, movilidad, sistema, embebidos, geolocalización, urbanismo**

**DISEÑO DE LA APLICACIÓN**

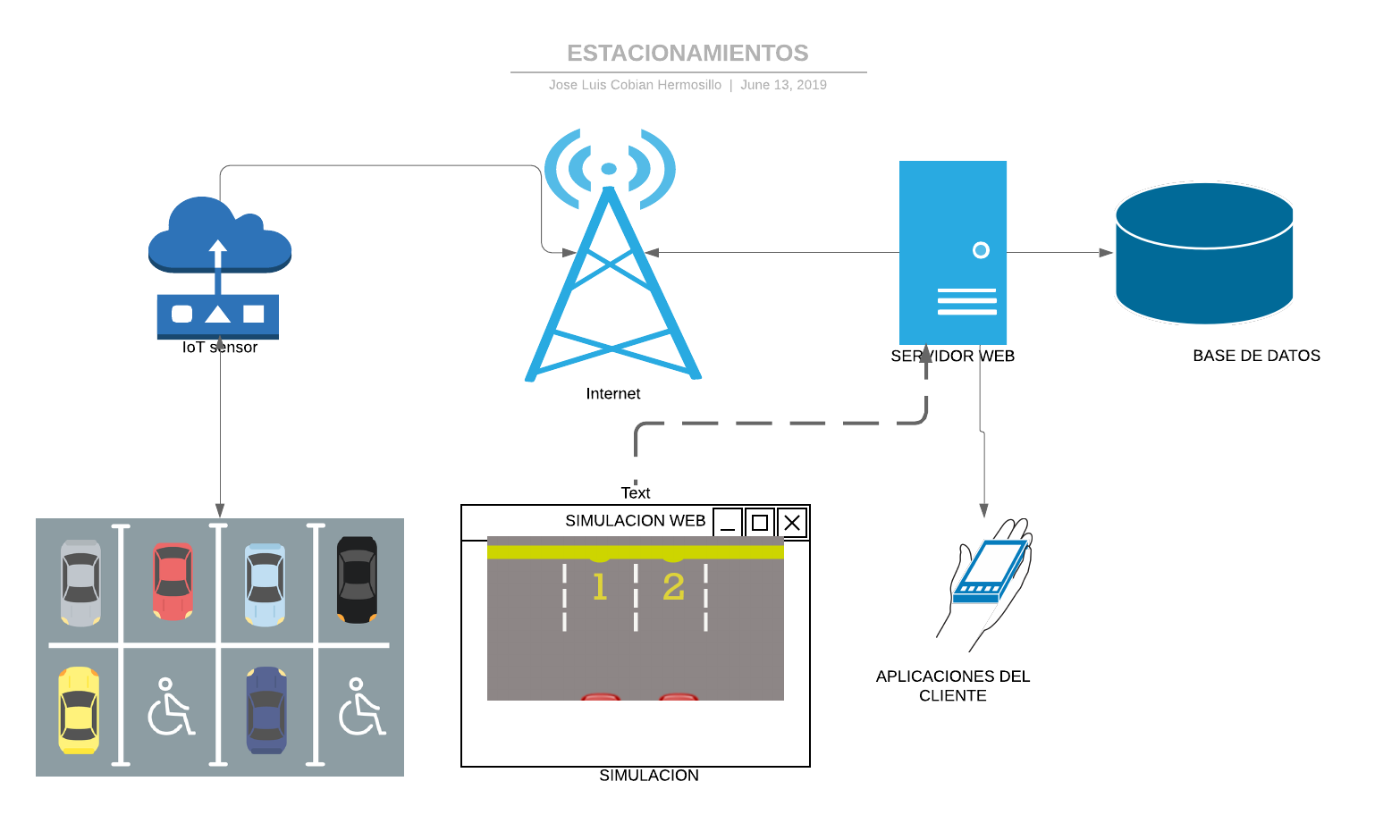
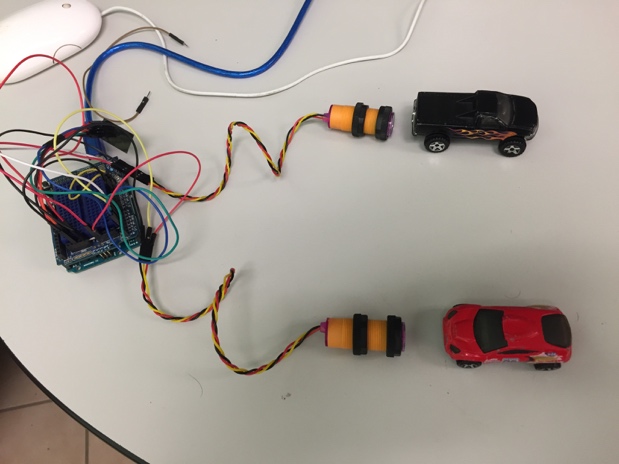


Figure 1. Esquema conceptual del proyecto

**FOTOS DEL PROTOTIPO**

** **

COMPONENTES

* Infrarojo e18
* Arduino Uno
* ESP8266-1

**ANEXOS**

**Codigo de la pagina**

Para poder funcionar tenemos que tener JQUERY

*estilo.css*

|  |
| --- |
| \*  Nombre de archivo: estilo.css  Luis Cobián  Estilos para proyecto IOT  1 de junio 2019 \*/  /\*  Estilos de tamaño del estacionamiento  borde y el fondo que se utilizo  \*/ #canvasAnimation {  width:800px;  height: 400px;  margin: 0 auto;  display: block;  border: 1px solid #d9d9d9;  background-image: url('../imgs/fondo.png') }   /\*  Formato para el encabezado  Centrado un poco mas abajo  \*/ h1 {  text-align:center; margin-top:20px } |

scriptIOT.js

|  |
| --- |
| */\*\*  \* Script del front para animación  \* Luis Cobián  \* 1 de Junio 2019  \*/* // variables del lienzo donde se manejara la animación var miCanvas, ctx; // variables que se necesitan para el primer cajon (auto ) var img1DespY, img1X, img1Y, ocupado1, interval1,animation1; // variables que se necesitan para el primer cajon (auto ) var img2DespY, img2X, img2Y, ocupado2, interval2, animation2; // variable de bloqueo ante las peticiones var lock = false;   //inicialización del canvas miCanvas = document.getElementById("canvasAnimation"); ctx = miCanvas.getContext("2d");  // inicializar con las imagenes var carImg1 = new Image(); carImg1.src="imgs/car.png"; // generar un duplicado del auto carImg2 = carImg1;  /// variables para los estados que tienen los cajones ocupado1 = false; ocupado2 = false; // posición de los dos autos img1X = 97; img2X = 165; img1Y = img2Y = 140;  /// Cuando la imagen se cargue se pinta. carImg1.onload = function(){  ctx.drawImage(carImg1,img1X,img1Y,46,90);  ctx.drawImage(carImg2,img2X,img2Y,46,90); }   // se manda llamar el loopMain para que empiece la animación setInterval(loopMain,3000)   /// Metodo cuando existe un cambio en el cajon 1 ya se ocupado o libre function ocupar1(){  if(!animation1){  if(ocupado1){  img1DespY = 1;  ocupado1 = false;  }else {  img1DespY = -1;  ocupado1 = true;  }  animation1=true;  interval1 = setInterval(moveCar1,20); // cada 20 milisegundos se ejecute  } }   // Metodo para animar el primer auto function moveCar1(){  ctx.clearRect(img1X,img1Y,46,90);  img1Y = img1Y + img1DespY;  ctx.drawImage(carImg1,img1X,img1Y,46,90);  if(img1Y<=28||img1Y>=140)  {  animation1=false;  clearInterval(interval1); // cancelar el ciclo  }  }   /// Metodo cuando existe un cambio en el cajon 1 ya se ocupado o libre function ocupar2(){  if(!animation2){  if(ocupado2){  img2DespY = 1;  ocupado2 = false;  }else {  img2DespY = -1;  ocupado2 = true;  }  animation2=true;  interval2 = setInterval(moveCar2,20);  } }  // Metodo para animar el primer auto function moveCar2(){  ctx.clearRect(img2X,img2Y,46,90);  img2Y = img2Y + img2DespY;  ctx.drawImage(carImg2,img2X,img2Y,46,90);  if(img2Y<=28||img2Y>=140)  {  animation2=false;  clearInterval(interval2);  }  }  /// metodo que se esta ejecutando infinitamente mientras se esta visualizando function loopMain() {  if(!lock && !animation1 && !animation2){  lock = true;  $.get("phps/refreshlugares.php",function(responseText,status){  try{  if(status=="success"){  res = JSON.parse(responseText);  res.forEach(function (lugar) {  switch (lugar.idLugares) {  case "1":  if(ocupado1!=parseInt(lugar.ocupado))  ocupar1();  break;  case "2":  if(ocupado2!=parseInt(lugar.ocupado))  ocupar2();  break;  }  });  }else{  alert(status);  }  }catch(e){  alert(e);  }  lock = false;  });  } } |

Conexion.php

|  |
| --- |
| <?php  */\*\*\*  \* Class Conexion  \* Genera la conexion con los datos prestablecidos en el archivo  \* Luis Cobián  \* 2 De junio  \*/* class Conexion {  private $host;  private $user;  private $pass;  private $db;  public $clave\_enc;   public function \_\_construct($d){  $this->host = $d['host'];  $this->user = $d['user'];  $this->pass = $d['pass'];  $this->db = $d['db'];  }   public function getConnection()  {   $conn = "";  try {  $conn = new PDO('mysql:host=' . $this->host . ';dbname=' . $this->db, $this->user, $this->pass);  $conn->setAttribute(PDO::*ATTR\_ERRMODE*, PDO::*ERRMODE\_EXCEPTION*);  } catch (PDOException $ex) {  echo $ex->getMessage();  $conn = $ex->getMessage();  }  return $conn;   } } |

connect.php

|  |
| --- |
| <?php  */\*\*\*  \* Datos  \* Luis Cobian  \* 2 de junio de 2019  \*/* $host = "localhost";  //$host = "127.0.0.1"; // comentar al poner en producción  $user = "luiscobi\_iotuser";  $pass = 'xxxxasassa;  $db = "luiscobi\_iot";   $conData = array('host' => $host,  'user' => $user,  'pass' => $pass,  'db' => $db  ); |

Lugares.php

|  |
| --- |
| <?php */\*\*\*  \* Class Lugares  \* Clase con las dos tareas que realizara el prototipo  \* Luis Cobiàn 2 de Junio 2019  \*/* class Lugares {  private $conn;   */\*\*  \* Lugares constructor.  \** ***@param*** *$conn datos de la conexion  \*/* public function \_\_construct($conn){  $this->conn = $conn;  }   */\*\*  \* Clase para consultar el estado de los estacionamientos  \** ***@return*** *retorna los datos o en su defecto el error que ocurrio  \*/* public function getEstadoLugares(){  $R['estado'] = "OK";  $c = $this->conn;  try {  $sql = $c->query("SELECT \* FROM Lugares");  $R['datos'] = $sql->fetchAll(PDO::*FETCH\_ASSOC*);  }catch(PDOExeption $e){  $R['estado'] = "ERROR: ".$e->getMessage();  }  return $R;  }   */\*\*  \* Metodo que permite cambiar el estado de los estacionamientos  \** ***@param*** *$idLugar id que identifica al lugar  \** ***@param*** *$ocupacion estado que manda el sensor  \** ***@return*** *regresa un valor de OK o en su defecto regresa el error  \*/* public function setOcupacion($idLugar, $ocupacion){  $R['estado'] = "OK";  $c = $this->conn;  try{  $sql = $c->prepare("UPDATE Lugares SET ocupado=:Ocupado WHERE idLugares=:Lugar");  $sql->execute(array('Lugar'=>$idLugar,  'Ocupado'=>$ocupacion));  $conn = null;  }catch(PDOExeption $e){  $R['estado'] = "ERROR: ".$e->getMessage();  }  return $R;  }   } |

cambioLugares.php

|  |
| --- |
| <?php $resultado['estado'] = "Error"; ///$datos = json\_decode(file\_get\_contents("php://input"));  if(isset($\_GET['lugar']) && isset($\_GET['estado'])) {   require("connect.php");  require("Conexion.php");  require("Lugares.php");   $conn = new Conexion($conData);   $c = new Lugares($conn->getConnection());  //$idlugar = (int)$datos->idLugar;  //$ocupacion = (int)$datos->ocupacion;   $idlugar = (int)$\_GET['lugar'];  $ocupacion = (int)$\_GET['estado'];   $file = fopen("archivo.txt", "w");   fwrite($file, "Datos: idLugar: " . $idlugar . " Ocupacion: " . $ocupacion . *PHP\_EOL*);  //fwrite($file, "Datos: idLugar: " . $datos->idLugar . " Ocupacion: " . $datos->ocupacion . PHP\_EOL);   fclose($file);   $resultado = $c->setOcupacion($idlugar, $ocupacion);   echo json\_encode($resultado); } |

refreshLugares.php

|  |
| --- |
| <?php */\*\*  \* Servicio web que recibe la peticiòn para la consulta de los datos  \* la pagina genera un JSON con los datos para ser consumidos  \*/* require("connect.php"); require("Conexion.php"); require("Lugares.php");  $conn = new Conexion($conData);  $c = new Lugares($conn->getConnection());  $res = $c->getEstadoLugares();  if($res['estado']=='OK')  echo json\_encode($res['datos']); else  echo json\_encode($res['estado']); |

Index.php

|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>Prototipo de aplicación IOT</title>  <link type="text/css" rel="stylesheet" href="css/estilo.css"/>   </head> <body>  <h1>ESTACIONAMIENTO X</h1>  <canvas id="canvasAnimation">  No se puede visualizar  </canvas>   <p><a href="https://github.com/cobymotion/PrototipoIOT">Github</a></p> </body>  <script src="js/jquery-3.4.1.min.js"></script> <script src="js/scriptIOT.js"></script>  </html> |

CODIGO DEL ARDUINO

|  |
| --- |
| /\*\*  \* Programa control para el envio de mensajes  \* basado en el sensor E18-D80NK  \* Sensor IR  \* Luis Cobián  \* 3 de junio  \*/  /\*\*  \* Constantes  \*/  #include <SoftwareSerial.h>  SoftwareSerial ESP(3, 2); // RX | TX  #define SENSOR1 4 // pin para conectar primer sensor  #define SENSOR2 7 // pin para conectar segundo sensor  #define INDICADOR1 12 // led indica primero cajon  #define INDICADOR2 13 // led indica segundo cajon  int estado1;  int estado2;  int lock1;  int lock2;  void setup() {  Serial.begin(9600);  ESP.begin(115200);  delay(1000);    pinMode(SENSOR1, INPUT\_PULLUP); // amarillo  pinMode(SENSOR2, INPUT\_PULLUP); // blanco  pinMode(INDICADOR1, OUTPUT);  pinMode(INDICADOR2, OUTPUT);  estado1=-1;  estado2=-1;  lock1=0;  lock2=0;  }  void loop() {  int left = digitalRead(SENSOR1);  int right = digitalRead(SENSOR2);  verificaSensor(left,INDICADOR1,&estado1,&lock1);  verificaSensor(right,INDICADOR2,&estado2,&lock2);    delay(500);  }  void verificaSensor(int valor, int indicador,  int \*estado, int \*lock) {  if(valor==0){  if(\*estado!=1) {  digitalWrite(indicador,HIGH);  \*estado=1;  \*lock=1;  imprimirEstado(estado);  }  else  \*lock=1;  }else {  if(\*estado!=0 && \*lock==0){  digitalWrite(indicador,LOW);  \*estado=0;  imprimirEstado(estado);  }  else  \*lock=0;  }      }  void imprimirEstado(int \*estado){  if(estado==&estado1)  {  Serial.print("Cajon1:");  Serial.println(estado1);  if(estado1==0)  enviarActualizacion("GET /iot/phps/cambioLugares.php?lugar=1&estado=0 HTTP/1.1");  else  enviarActualizacion("GET /iot/phps/cambioLugares.php?lugar=1&estado=1 HTTP/1.1");  }  else {  Serial.print("Cajon2:");  Serial.println(estado2);  if(estado2==0)  enviarActualizacion("GET /iot/phps/cambioLugares.php?lugar=2&estado=0 HTTP/1.1");  else  enviarActualizacion("GET /iot/phps/cambioLugares.php?lugar=2&estado=1 HTTP/1.1");  }  }  void enviarActualizacion(char \*strHTTP){  Serial.println("Enviar actualización");  Serial.println(strHTTP);  ESP.println("AT+CIPSTART=\"TCP\",\"luiscobian.com.mx\",80");  delay(200);  ESP.println("AT+CIPSEND=102");  delay(200);  ESP.println(strHTTP);  //ESP.println("GET /iot/phps/cambioLugares.php?lugar=1&estado=0 HTTP/1.1");  delay(200);  ESP.println("Host:luiscobian.com.mx");  delay(200);  ESP.println("Conexion: close");  delay(200);  ESP.println("");  ESP.println("AT+CIPCLOSE");  delay(2500);  } |

**BIBLIOGRAFIA**

<https://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/>

<https://naylampmechatronics.com/sensores-proximidad/236-sensor-de-proximidad-fotoelectrico-infrarrojo-e18-d80nk.html>

<https://www.hackster.io/Manoranjan2050/dht11-and-nodemcu-with-blynk-10e6b>

<https://www.php.net/manual/es/reserved.variables.get.php>

<https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/client-examples.html>