



Petfy

El guardian de tu mascota



¿Que es petfy?

Es un dispensador de alimento inteligente para tu mascota, desarrollado con las ultimas tecnologías, Petfy se conecta a la nube por lo que te permite cuidar de tu mascota de manera remota, en cualquier lugar que te encuentres.

Puedes programar de manera muy fácil las comidas y porciones que tu perro necesita, a cualquier hora y todo en tiempo real.

Petfy te notifica cuando la comida de contenedor este próxima a terminarse, así como también cuando tu mascota se haya terminado su porción.



Objetivo

Implementar una solución a un problema cotidiano, en lo que respecta a la alimentación de mascotas (perros o gatos) mediante un alimentador automático. Que permita controlar desde un dispositivo móvil la hora y porción de cada alimento.

Justificación

Tener una mascota representa una gran responsabilidad, en la actualidad por cuestiones de trabajo y tiempo de cada persona, resulta complicado ocuparse adecuadamente de la alimentación de sus mascotas. Es por ello que pensamos que mejorar un alimentador automático podría contribuir en mejorar la calidad de alimentación y cuidado de las mascotas.



Material

- 2 Sensores Ultra sonico.
- 1 Servomotor.
- 1 Particle IO.
- 1 Contenedor
- 1 Plato de comida

Descripción

Capacidad: 2kg.
Capacidad plato: 400gramos.
Alimentación: 120vca.
Dimensiones: 38cm X 30cm X 50cm.
WIFI

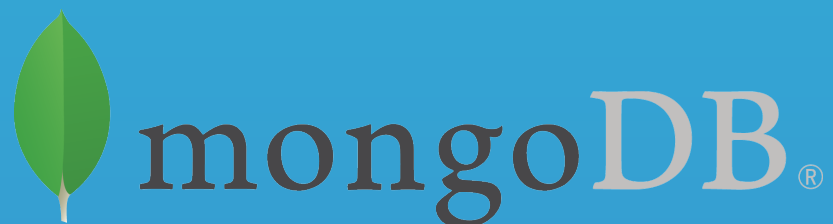


Tecnologías

Petfy esta desarrollado con las ultimas tecnologías para desarrollo de software y hardware, por lo que tendrás la confianza de un producto estable y duradero.

Servidor

Desarrollado en JAVASCRIPT con MEAN stack. (MongoDB, Express, Angular y NodeJS). Frameworks para manejo de BigData y servidores escalables.





Tecnologías

Cliente (Aplicación)

Aplicación desarrollada con código nativo, Xcode con SWIFT para la plataforma iOS y ANDROID STUDIO con JAVA para la plataforma ANDROID.

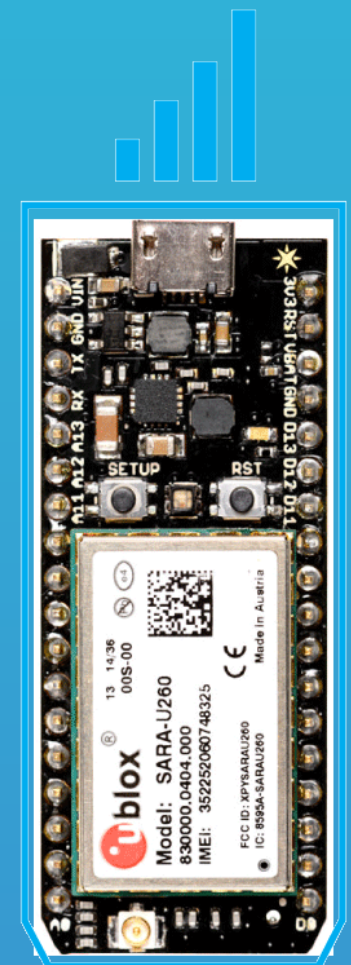
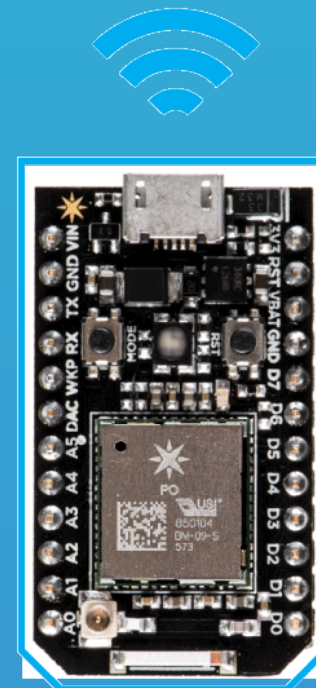
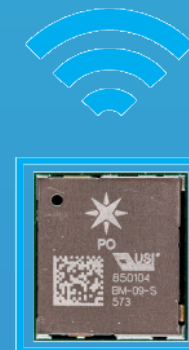




Tecnologías

Hardware

Particle IO, plataforma para desarrollo de hardware a nivel celular, y escalable. Cuenta con una API propia para la suscripción de métodos y publicación de variables





Evidencias

Xcode interface showing the Petfy app development environment. The interface includes a menu bar, a toolbar, a project navigator, a document navigator, and a main canvas area.

The project navigator on the left shows the Petfy project structure, including files like AppDelegate.swift, ViewController.swift, DashboardViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, and Info.plist. The document navigator on the right shows the selected storyboard, Main.storyboard (Base), and the selected scene, Dashboard View Controller Scene.

The main canvas area displays two storyboard scenes connected by a segue. The first scene, titled "View Controller", shows a "Log In" screen with a dog illustration and a text input field. The second scene, titled "Dashboard View Controller", shows a dashboard with a "Log Out" button, a "Petfy" header, a dog illustration, "Owner Name" and "Cloud Petfy" labels, a "Estado actual" section with "Nivel de comida" and "Comida en plato" both at 80%, a "Configurar porción" section with a "Porción:" label, a value of 1, and buttons for "Minutos" and "Horas", and a "Guardar" button. The bottom of the dashboard has a "Servir alimento" button.

The status bar at the bottom indicates the device is an iPhone 7 (wC hR) and the zoom level is 78%.



Evidencias

Atom File Edit View Selection Find Packages Window Help

server.js — ~/Documents/PROJECTS/PERSONAL/PETFY/MEAN/Petfy_Admin

Petfy_Admin

- .git
- app
- bower_components
- config
- node_modules
- public
 - images
 - js
 - controllers
 - admin
 - dashboardController.js
 - productsController.js
 - servicesController.js
 - usersController.js
 - index
 - .DS_Store
 - routes
 - services
 - admin.js
 - index.js
 - login.js
 - pages
 - styles
 - .DS_Store
 - .gitignore
 - admin.js
 - bower.json
 - ecosystem.json
 - gulpfile.js
 - package.json
 - README.md
 - server.js

```
1
2 var express = require("express");
3 var app = express();
4
5 var bodyParser = require("body-parser");
6 var morgan = require("morgan");
7
8 //Config File
9 process.env.NODE_ENV = process.env.NODE_ENV || 'test';
10
11 var config = require('./config/config');
12 var port = process.env.PORT || config.port;
13
14 //APP CONFIGURATION:
15 function headers(req,res,next){
16   res.setHeader('Access-Control-Allow-Origin', '*');
17   res.setHeader('Access-Control-Allow-Methods', 'GET,POST');
18   res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type, \ authorization');
19   next();
20 }
21
22 app.use(bodyParser.urlencoded({extended:true}));
23 //app.use(bodyParser.json());
24 app.use(headers);
25 app.use(morgan("dev"));
26
27 //SET THE ROUTES OF ALL:
28 require("./app/routes/setroutes")(__dirname,app,express);
29
30 //START SERVER:
31 app.listen(port);
32 console.log("Server started at port: " + port);
33
```

File 0 Project 0 No Issues server.js 1:1

LF UTF-8 Babel ES6 JavaScript master 2 updates



Evidencias

```
//
//  ViewController.swift
//  PetFy
//
//  Created by Jose De Jesus Garfias Lopez on 12/17/16.
//  Copyright © 2016 Karmapulse. All rights reserved.
//

import UIKit
import Firebase
import GoogleSignIn

class ViewController: UIViewController, GIDSignInUIDelegate, GIDSignInDelegate {

    var googleProfile: GIDProfileData!;
    var fireProfile:FIRUser!;

    @IBOutlet weak var SignIn: GIDSignInButton!

    override func viewDidLoad() {
        super.viewDidLoad()

        GIDSignIn.sharedInstance().delegate = self;
        GIDSignIn.sharedInstance().uiDelegate = self

        let gradient: CAGradientLayer = CAGradientLayer()
        gradient.colors = [
            UIColor(red: 41/255, green: 201/255, blue: 221/255, alpha: 1).CGColor,
            UIColor(red: 56/255, green: 150/255, blue: 207/255, alpha: 1).CGColor ];
        gradient.locations = [0.0 , 1.0]
        gradient.frame = CGRect(x: 0.0, y: 0.0, width: self.view.frame.size.width, height: self.view.frame.size.height);

        self.view.layer.insertSublayer(gradient, at: 0);
    }

    func alerta(_ Titulo:String,Mensaje:String){
        let alertController = UIAlertController(title: Titulo, message:
        Mensaje, preferredStyle: UIAlertControllerStyle.alert);
        let okAction = UIAlertAction(title: "OK", style: UIAlertActionStyle.default) {
            UIAlertAction in
            print("Ok Pressed");
        }
        alertController.addAction(okAction);
        self.present(alertController, animated: true, completion: nil)
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if(segue.identifier == "DashboardSegue"){
```



Evidencias

Safari File Edit View History Bookmarks Develop Window Help

build.particle.io

petfy.ino rest_client.cpp rest_client.h

```
1
2 // This #include statement was automatically added by the Particle IDE.
3 #include "rest_client.h"
4 // This #include statement was automatically added by the Particle IDE.
5 #include "HC_SR04/HC_SR04.h"
6
7 // Global vars
8 double bucketDistance = 0;
9 double dishDistance = 0;
10
11 bool notifiedBucket = false;
12 bool notifiedDish = false;
13
14 int bucketTrigger = A0;
15 int bucketEcho = A1;
16 int dishTrigger = A2;
17 int dishEcho = A3;
18 int stateLed = A4;
19 int bucketLed = A5;
20 int dishLed = A6;
21
22 int portion = 1;
23 int unitTime = 0;
24 String tokenForPush = "";
25
26 // Expose Functions
27 int serveFood(String command);
28 int setPushToken(String token);
29 int setPortion(String newPortion);
30 int setUnitTime(String newUnitTime);
31
32 // Class Intances
33 Servo Door;
34 HC_SR04 bucketFinder = HC_SR04(bucketTrigger, bucketEcho);
35 HC_SR04 dishFinder = HC_SR04(dishTrigger, dishEcho);
36 RestClient client = RestClient("fcm.googleapis.com");
37
38 void setup()
39 {
40
41     Spark.function("serveFood", serveFood);
42     Spark.function("setPushToken", setPushToken);
43     Spark.function("setPortion", setPortion);
44     Spark.function("setUnitTime", setUnitTime);
45
46     Spark.variable("portion", &portion, INT);
47     Spark.variable("unitTime", &unitTime, INT);

```

Ready.

Particle Apps

Current App

PETFY

Optional description

Files

PETFY.INO

REST_CLIENT.CPP

REST_CLIENT.H

Included libraries

HC_SR04

SPARKCOREPOLLEDTIMER

REMOVE APP

My apps

Type to find

PETFY

CREATE NEW APP

Example apps

1. BLINK AN LED
2. WEB-CONNECTED LED
3. FUNCTION VARIABLE
4. PUBLISH
5. SUBSCRIBE



Descarga de códigos fuente



<https://dl.dropboxusercontent.com/u/83047324/PETFY.zip>