PERÍODO 2020

A picture containing food, drawing

Description automatically generated

**MANUAL TÉCNICO DEL PROYECTO**

Tema 6: Aplicación del monitoreo de productos en la refrigeradora

**Grupo #1**

**INTEGRANTES:**

* ELVIA DANIELA DE LA CRUZ INTRIAGO
* CARLOS AXXEL NARANJO RIOFRIO
* SERGIO KEVIN VÉLIZ ALDAZ
* FRANK ANDRES VERA VERA

# **Índice**

[Índice 1](#_Toc50074519)

[1. Resumen ejecutivo 2](#_Toc50074520)

[2. Descripción del problema 2](#_Toc50074521)

[3. Objetivos específicos 3](#_Toc50074522)

[4. ¿Cómo funciona la solución? 4](#_Toc50074523)

[5. ¿Qué van a construir para resolver el problema? 4](#_Toc50074524)

[6. Recursos de hardware y de software 5](#_Toc50074525)

[7. Explicación paso a paso de la implementación del proyecto 7](#_Toc50074526)

[8. Diagramas de diseño del proyecto, diagrama de circuito, diagrama del modelo entidad-relación, diagrama de casos UML, diagrama de despliegue 8](#_Toc50074527)

[9. Descripción de los campos, tipos de datos creados en la base de datos y el código SQL 12](#_Toc50074528)

[10. Explicación del código fuente completo desarrollado con los comentarios correspondientes 24](#_Toc50074529)

[11. Análisis de presupuesto 44](#_Toc50074530)

[12. Conclusiones 45](#_Toc50074531)

[13. Referencias bibliográficas 46](#_Toc50074532)

# **Resumen ejecutivo**

El presente proyecto consiste en la automatización de un refrigerador. Se desarrollará un aplicativo web que permita al usuario llevar un control de su despensa en el refrigerador, conocer qué alimentos almacena y la cantidad que dispone.

En este contexto, se agruparán los alimentos en distintos grupos como vegetales, hortalizas, frutas, lácteos, carnes, etc. Se destinará un espacio para cada categoría y mediante celdas de carga ubicadas en cada uno de dichos espacios se censarán los respectivos pesos para llevar un recuento de estos y poder reportarlo periódicamente al usuario.

# **Descripción del problema**

Según la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) “Aproximadamente un tercio de los alimentos que son producidos en el mundo terminan siendo desechados” [[1]](#footnote-1)

Un habitual problema en la administración de los hogares es la cantidad de comida que se pierde. Muchas veces no se tiene conocimiento de todos los alimentos almacenados en el refrigerador, lo que desencadena la compra innecesaria de los mismos productos que inicialmente ya se tenían; y debido a la falta de organización y alta cantidad de productos que se llegan a tener, parte de ellos vencen sin ser consumidos teniendo que ser desechados.

Se requiere automatizar el hogar de un usuario, en el caso específico de este proyecto, se necesita la automatización de su refrigerador. Para ello, se solicita llevar a cabo el desarrollo de un aplicativo web que le permita al usuario revisar continuamente el contenido de su refrigerador, conocer los alimentos de los cuales dispone y las cantidades que posee de los mismos.

El usuario a podrá mediante consultas en la página web de manera periódica estar al tanto de las guarniciones que posee y en el caso de ya no contar con determinados productos poder reponerlos.

# **Objetivos específicos**

* Diseñar un aplicativo web que permita al usuario revisar información acerca de sus guarniciones en el refrigerador desde cualquier parte.
* Informar al usuario cuando ya no cuente con determinados alimentos en su refrigerador.
* Combinar el uso de hardware, bases de datos y aplicativos web para el diseño de una solución que facilite al usuario el control de alimentos en su refrigerador.

# **¿Cómo funciona la solución?**

Para la automatización del refrigerador, se ha propuesto utilizar sensores de peso para medir la cantidad de alimentos en el refrigerador; esto funcionará primero, delimitándole a la refrigeradora espacios específicos para cada alimento, donde también se seleccionará un peso máximo por alimento, por ejemplo se colocaran en un contenedor solo manzana, en otro solo carnes, otro para yogurt y así respectivamente, luego se coloca una celda de carga para cada contenedor, y se conectan las celdas de carga a un controlador, que traducirá la señal obtenida de la celda en un peso y un porcentaje para el envío a la nube.

Luego esta información es enviada a un gateway, que permite el envío de la información a la red, donde puede será manejada por un hoster como webhost para el manejo de la base de datos y la presentación en la página web.

# **¿Qué van a construir para resolver el problema?**

A nivel de hardware se llevará a cabo la construcción de dos bloques, uno de medición y otro destinado a la comunicación, estos se detallan a continuación:

* Bloque de medición: Aquí se encontrará el sensor a utilizar y el Arduino. El sensor consiste en una celda de carga de 50kg, la cual se comunica mandando señales análogas, este sensor es conectado al módulo hx711 que traduce esa señal análoga a digital de 24 bits para su mejor interpretación. Este módulo será el que se conecte con el Arduino para ejecutar cualquier tipo de decisión que se quiera con esos datos, en este caso su función principal es leer la señal del módulo y actualizarlos en la base de datos.
* Bloque de comunicación: Este consta de la incorporación modular de una Ethernet Shield W5100 al Arduino para permitir la conexión de este último con la red para el envió de información a la base de datos mediante peticiones html.

En cuanto al modelo físico a construir, se diseñarán los contenedores para los grupos de alimentos que tendrán en su base cada uno una celda de carga, esto se ubicará dentro del refrigerador para el almacenamiento de guarniciones. Las celdas de carga a su vez se conectarán al resto de los elementos detallados anteriormente, los cuales se encontrarán externos al refrigerador para evitar fallas en su funcionamiento debido a las condiciones térmicas y conectados a una toma de energía de la casa en caso de requerir alimentación.

# **Recursos de hardware y de software**

* Recursos de hardware utilizados
  + 1 Arduino UNO: Se decidió utilizar esta placa de microcontrolador porque es de código abierto y ya se tienen conocimiento de su manejo y funcionamiento al haber sido utilizada en otros proyectos por los integrantes del grupo. Cabe mencionar que en el caso de que el usuario requiera colocar un gran número de contenedores en el refrigerador se deberá utilizar una versión de Arduino con un mayor número de pines como el Mega para la conexión de todas las celdas de carga.
  + Celdas de carga 50kg: Estos sensores funcionarán como un transductor que puede medir el peso o la fuerza aplicada en la celda y la convierte en una señal eléctrica para ser pasada al Arduino. Su rango de temperatura de operación es de 0 a 50°C, ideal para los refrigeradores que operan generalmente entre 3 y 4°C.
  + Módulo hx711: Este es un transmisor para celdas de carga, será utilizado ya que permite obtener lecturas con buena precisión y confiables de las celdas.
  + Ethernet Shield W5100: Este controlador de Ethernet será utilizado pues facilita la implementación de conectividad a internet sin necesidad de un SO, por lo que es indicado para aplicaciones de IoT. (Llamas, 2017) Permitirá la conexión del Arduino a la red para el envio de datos recolectados con las celdas de carga.
* Recursos de software que se utilizará.
  + Arduino: Es una interfaz (IDE) de programación que usa un lenguaje basado en C++, en la actualidad es uno de los IDE más famoso en aplicación debido a su facilidad de programación.

Este software fue utilizado para poder programar y controlar los datos receptados por la celda de carga, también para mandar información hacia la base de datos en la red.

* + MySQL: Es un sistema de gestión de bases de datos relacional de código abierto, siendo una base datos de muy rápida lectura, su uso tiene gran popularidad en aplicaciones de redes sociales muy conocidas, generalmente su funcionalidad se ve ligada con PHP.

En nuestro proyecto usamos MySQL para guardar y tabular la información obtenida a través de la celda de carga. También se utilizará para guardar los datos de registro de los usuarios como nombre, correo, etc.   
Una de las ventajas con este software es la rápida actualización de los datos.

* + PHP: Es un lenguaje de programación diseñado especialmente para desarrollo web, programaremos MySQL con PHP, es una combinación habitual en muchos proyectos, para poder tener una mejor lectura y tabulación de los datos obtenidos. Mediante este lenguaje se llevará a cabo la asociación de la basa de datos con la página web donde finalmente se mostrará al usuario la información obtenida acerca de sus guarniciones en refrigerador.

# **Explicación paso a paso de la implementación del proyecto**

Se creó una cuenta en el web hosting “000WebHost” de tipo gratuita con uso exclusivo destinado para el presente proyecto. Luego de diseñar el diagrama entidad-relación a utilizar, se procedió a crear una base de datos.

Posteriormente, se llevó a cabo el diseño de una página web donde el usuario pueda registrarse, iniciar sesión y acceder a la información acerca de los alimentos almacenados en su refrigerador. Todo esto, haciendo uso de phph, html y css para brindar una interfaz fácil de utilizar y agradable a la vista del usuario. Finalmente, la base de datos fue asociada la página web.

En cuanto a la parte física o adaptaciones realizadas en el refrigerador se tienen los siguientes pasos:

* Diseño de los contenedores en los cuales se agruparán los distintos alimentos en el refrigerador, estos deben tener un modelo estándar funcional para distintos modelos de refrigeradores.
* Añadir a la base de cada contenedor una celda de carga.
* Conectar las celdas de carga a una caja externa del refrigerador que contendrá el circuito formado por los controladores para las celdas, el Arduino UNO y la Ethernet Shield W5100.
* Se debe ingresar a la tarjeta Arduino el código creado para la recepción de datos de las celdas de carga y el direccionamiento de estos a la base de datos.
* Alimentar la tarjeta Arduino.

# **Diagramas de diseño del proyecto, diagrama de circuito, diagrama del modelo entidad-relación, diagrama de casos UML, diagrama de despliegue**

Diagrama de diseño del proyecto

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Imagen 1 Diagrama de diseño del proyecto – vista general

Imagen que contiene refrigerador

Descripción generada automáticamente

Imagen 2 Diagrama de diseño del proyecto - vista interna del refrigerador donde se encontrarán los contenedores con sus respectivas celdas de carga que irán contactadas al resto del circuito ubicado en el exterior del refrigerador

Diagrama del circuito

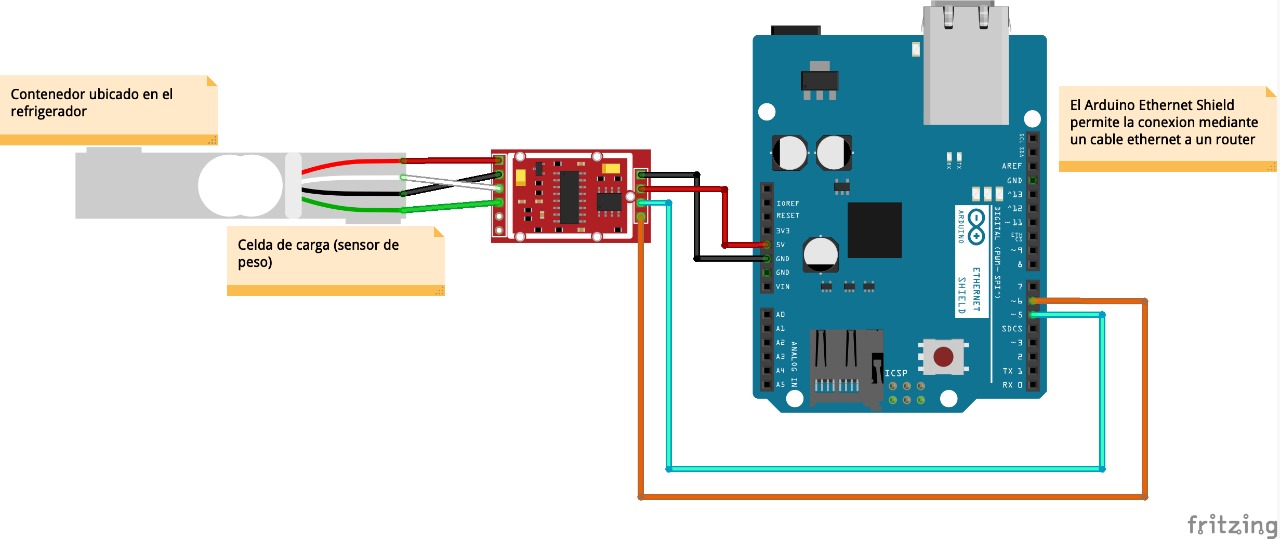


Imagen 3 Diagrama del circuito, conexión de una celda de carga (El número de celdas de carga a conectar dependerá del número de contenedores que se deseen instalar en el refrigerador)

Diagrama del modelo entidad-relación

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Imagen 4 Diagrama del modelo entidad-relación

Diagrama de casos UML

Imagen que contiene texto, mapa

Descripción generada automáticamente

Imagen 5 Diagrama de casos UML

Diagrama de despliegue

Captura de pantalla de un celular

Descripción generada automáticamente

Imagen 6 Diagrama de despliegue

# **Descripción de los campos, tipos de datos creados en la base de datos y el código SQL**

Descripción de los campos y tipos de datos creados en la base de datos

* Usuario: Almacenará la información del registro de usuario, necesaria para futuros inicios de sesión
  + id (PK): dato único para cada usuario.
  + nombre: Almacena el nombre completo del usuario
  + contrasenia: Almacena contraseña ingresada por usuario en su registro
  + correo: Almacena correo de usuario, requerido para enviar futuras alertas y actualizaciones referentes al contenido de alimentos en su refrigerador
  + usuario: Almacena el usuario ingresado en el registro, con el cuál se llevará a cabo el inicio de sesión.
* Refrigerador: Permitirá la asociación del usuario con su refrigerador
  + id (PK): dato único para cada refrigerador
  + idUsuario (FK): guarda el id del usuario propietario del refrigerador
* Sensor: Requerido para identificar a los sensores ubicados en cada contenedor, desde los cuales llegarán los datos de pesos de alimentos almacenados.
  + id (PK): dato único para reconocer a cada sensor
  + pesoMax: establecerá el peso máximo que aguanta la celda de carga para su funcionamiento óptimo.
* Registro-Alimento: Permitirá el manejo de la relación de muchos a muchos entre los refrigeradores desde los cuales se solicita el registro de nuevos datos recopilados y los tipos de alimentos a los cuales referirán las actualizaciones de peso.
  + id: (PK) dato único para cada Registro-Alimento
  + idSensor (PK):almacena el id del sensor desde el cuál se dirige la información de peso.
  + idRefri (PK): almacena el id del refrigerador que contiene al sensor desde el cual se dirige la actualización de peso.
  + peso: almacena el dato obtenido por el sensor acerca del peso actual en alimentos que tiene un determinado contenedor.
  + fecha: fecha en la cuál se está realizando la actualización o ingreso de nuevo peso
  + hora: hora en la cuál se está realizando la actualización o ingreso de nuevo peso
* Tipo\_alimento: categorías en las cuáles se agruparán los distintos alimentos
  + id (PK): Dato único para identificar cada tipo de alimento
  + nomTip: Almacena el tipo de alimento. Ej: Frutas, Vegetales, Carnes, etc.
* Alimento: requerido para especificar el alimento almacenado en cada contenedor y sensado por cada celda de carga
  + nombre (PK): Almacena el nombre específico del alimento
  + idTipo (FK): Permitirá la asignación de agrupación del alimento a una de las clasificaciones (Lácteos, Frutas, Vegetales, etc)
  + pesoMin: almacena el peso mínimo de alimento que se desea tener, cuando el peso obtenido por sensor sea inferior a este, se procederá a alertar al usuario.
  + pesoMax: almacena el peso de alimento máximo o requerido por el usuario, en base a este el peso obtenido por sensor se podrá mostrar también como un porcentaje de alimento del cual se dispone.
* Sensor-Alimento: Permite relacionar un sensor (celda de carga) con el alimento del cuál se encontrará realizando las mediciones de peso
  + Id (PK): dato único para hacer referencia a la relación sensor-alimento
  + idSensor: almacena el id del sensor en cuestión.
  + idAlimento: hace referencia al tipo de alimento del cuál se está obteniendo el peso.

Código SQL

EL código mostrado a continuación fue exportado desde phpmyadmin:

-- phpMyAdmin SQL Dump

-- version 4.9.5

-- https://www.phpmyadmin.net/

--

-- Servidor: localhost:3306

-- Tiempo de generación: 04-09-2020 a las 18:13:44

-- Versión del servidor: 10.3.16-MariaDB

-- Versión de PHP: 7.3.12

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

--

-- Base de datos: `id14446876\_refrigeradoriot`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Alimento`

--

CREATE TABLE `Alimento` (

`nombre` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`estado` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`idTipo` int(11) NOT NULL,

`pesoMin` float NOT NULL,

`pesoMax` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Alimento`

--

INSERT INTO `Alimento` (`nombre`, `estado`, `idTipo`, `pesoMin`, `pesoMax`) VALUES

('Agua', 'Liquido', 5, 0.2, 1.5),

('Cebolla', 'Solido', 2, 0.125, 0.625),

('Cerdo', 'Solido', 3, 0.06, 1.5),

('Durazno', 'Solido', 1, 0.117, 1.17),

('Guineo', 'Solido', 1, 0.15, 1.5),

('Huevos', 'Solido', 6, 0.05, 0.6),

('Jugo', 'Liquido', 5, 0.3, 1.8),

('Leche', 'Liquido', 4, 0.24, 3),

('Pimiento', 'Solido', 2, 0.085, 0.425),

('Pollo', 'Solido', 3, 0.845, 3.38),

('Queso', 'Solido', 4, 0.03, 1),

('Res', 'Solido', 3, 0.045, 1.5),

('Tomate', 'Solido', 2, 0.2, 1),

('Zanahoria', 'Solido', 2, 0.075, 0.375);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Refrigerador`

--

CREATE TABLE `Refrigerador` (

`id` int(11) NOT NULL,

`idUsuario` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Refrigerador`

--

INSERT INTO `Refrigerador` (`id`, `idUsuario`) VALUES

(1, 2);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Registro-Alimento`

--

CREATE TABLE `Registro-Alimento` (

`id` int(11) NOT NULL,

`idSensor` int(11) NOT NULL,

`idRefri` int(11) NOT NULL,

`peso` int(11) NOT NULL,

`fecha` date NOT NULL,

`hora` time NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Registro-Alimento`

--

INSERT INTO `Registro-Alimento` (`id`, `idSensor`, `idRefri`, `peso`, `fecha`, `hora`) VALUES

(1, 1, 1, 1, '2020-08-31', '02:40:59'),

(2, 2, 1, 0, '2020-08-31', '02:40:59'),

(3, 1, 1, 1, '2020-09-01', '02:42:12'),

(4, 2, 1, 0, '2020-09-01', '02:42:12'),

(5, 1, 1, 2, '2020-09-04', '08:31:44'),

(6, 1, 1, 3, '2020-09-04', '08:33:19'),

(7, 1, 1, 0, '2020-09-04', '08:33:35');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Sensor`

--

CREATE TABLE `Sensor` (

`id` int(11) NOT NULL,

`pesoMax` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Sensor`

--

INSERT INTO `Sensor` (`id`, `pesoMax`) VALUES

(1, 50),

(2, 50),

(3, 50),

(4, 50),

(5, 50),

(6, 50),

(7, 50),

(8, 50),

(9, 50),

(10, 50),

(11, 50),

(12, 50),

(13, 50),

(14, 50);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Sensor-Alimento`

--

CREATE TABLE `Sensor-Alimento` (

`id` int(11) NOT NULL,

`idSensor` int(11) NOT NULL,

`idAlimento` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Sensor-Alimento`

--

INSERT INTO `Sensor-Alimento` (`id`, `idSensor`, `idAlimento`) VALUES

(1, 1, 'Cerdo'),

(2, 2, 'Res'),

(3, 3, 'Pollo'),

(4, 4, 'Durazno'),

(5, 5, 'Guineo'),

(6, 6, 'Cebolla'),

(7, 7, 'Pimiento'),

(8, 8, 'Tomate'),

(9, 9, 'Zanahoria'),

(10, 10, 'Leche'),

(11, 11, 'Queso'),

(12, 12, 'Huevos'),

(13, 13, 'Agua'),

(14, 14, 'Jugo');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Tipo\_alimento`

--

CREATE TABLE `Tipo\_alimento` (

`id` int(11) NOT NULL,

`nomTipo` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Tipo\_alimento`

--

INSERT INTO `Tipo\_alimento` (`id`, `nomTipo`) VALUES

(1, 'Frutas'),

(2, 'Vegetales'),

(3, 'Carnes'),

(4, 'Lacteos'),

(5, 'Bebidas'),

(6, 'Otros');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Usuario`

--

CREATE TABLE `Usuario` (

`id` int(11) NOT NULL,

`nombre` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`contrasenia` varchar(25) COLLATE utf8\_unicode\_ci NOT NULL,

`correo` varchar(30) COLLATE utf8\_unicode\_ci NOT NULL,

`usuario` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Volcado de datos para la tabla `Usuario`

--

INSERT INTO `Usuario` (`id`, `nombre`, `contrasenia`, `correo`, `usuario`) VALUES

(1, 'Sergio Veliz', 'contrasenia', 'skveliz@espol.edu.ec', 'Sergio'),

(2, 'pstnombre', 'contrasenia', 'pst20201T@outlook.com', 'pstusuario');

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `Alimento`

--

ALTER TABLE `Alimento`

ADD PRIMARY KEY (`nombre`),

ADD KEY `idTipo` (`idTipo`);

--

-- Indices de la tabla `Refrigerador`

--

ALTER TABLE `Refrigerador`

ADD PRIMARY KEY (`id`),

ADD KEY `idUsuario` (`idUsuario`);

--

-- Indices de la tabla `Registro-Alimento`

--

ALTER TABLE `Registro-Alimento`

ADD PRIMARY KEY (`id`),

ADD KEY `idSensor` (`idSensor`,`idRefri`),

ADD KEY `idRefri` (`idRefri`);

--

-- Indices de la tabla `Sensor`

--

ALTER TABLE `Sensor`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `Sensor-Alimento`

--

ALTER TABLE `Sensor-Alimento`

ADD PRIMARY KEY (`id`),

ADD KEY `idSensor` (`idSensor`),

ADD KEY `idAlimento` (`idAlimento`);

--

-- Indices de la tabla `Tipo\_alimento`

--

ALTER TABLE `Tipo\_alimento`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `Usuario`

--

ALTER TABLE `Usuario`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `Refrigerador`

--

ALTER TABLE `Refrigerador`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT de la tabla `Registro-Alimento`

--

ALTER TABLE `Registro-Alimento`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT de la tabla `Sensor`

--

ALTER TABLE `Sensor`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT de la tabla `Sensor-Alimento`

--

ALTER TABLE `Sensor-Alimento`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT de la tabla `Tipo\_alimento`

--

ALTER TABLE `Tipo\_alimento`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT de la tabla `Usuario`

--

ALTER TABLE `Usuario`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `Alimento`

--

ALTER TABLE `Alimento`

ADD CONSTRAINT `Alimento\_ibfk\_1` FOREIGN KEY (`idTipo`) REFERENCES `Tipo\_alimento` (`id`) ON UPDATE CASCADE;

--

-- Filtros para la tabla `Refrigerador`

--

ALTER TABLE `Refrigerador`

ADD CONSTRAINT `Refrigerador\_ibfk\_1` FOREIGN KEY (`idUsuario`) REFERENCES `Usuario` (`id`) ON UPDATE CASCADE;

--

-- Filtros para la tabla `Registro-Alimento`

--

ALTER TABLE `Registro-Alimento`

ADD CONSTRAINT `Registro-Alimento\_ibfk\_1` FOREIGN KEY (`idSensor`) REFERENCES `Sensor` (`id`) ON UPDATE CASCADE,

ADD CONSTRAINT `Registro-Alimento\_ibfk\_2` FOREIGN KEY (`idRefri`) REFERENCES `Refrigerador` (`id`) ON UPDATE CASCADE;

--

-- Filtros para la tabla `Sensor-Alimento`

--

ALTER TABLE `Sensor-Alimento`

ADD CONSTRAINT `Sensor-Alimento\_ibfk\_1` FOREIGN KEY (`idSensor`) REFERENCES `Sensor` (`id`) ON UPDATE CASCADE,

ADD CONSTRAINT `Sensor-Alimento\_ibfk\_2` FOREIGN KEY (`idAlimento`) REFERENCES `Alimento` (`nombre`) ON UPDATE CASCADE;

COMMIT;

# **Explicación del código fuente completo desarrollado con los comentarios correspondientes**

**Código ingresado a tarjeta Arduino**

#include <HX711.h>

#include <SPI.h>

#include <Ethernet.h>

#define calibration\_factor -7050.0 //Valor de calibración obtenido de internet, para aplicaciones reales se debe obtener con la celda de carga

#define DOUT 3

#define CLK 2

float peso;

HX711 scale;

//Dirección mac, ip, gateway y mascara para conexión

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress server(192, 168, 1, 6);

EthernetClient cliente;

void setup() {

Serial.begin(9600);

//Inicializando la balanza

scale.begin(DOUT, CLK);

scale.set\_scale(calibration\_factor);

scale.tare(); //Se encera la celda de carga al iniciar la instalacion del sensor

peso = 0;

//Inicializando la conexion

if (Ethernet.begin(mac) == 0) {

Serial.println("Fallo en la conexion con Ethernet");

}

if (cliente.connect(server, 80)) {

Serial.println("Conectado");

}

else {

Serial.println("Conexion Fallida");

}

}

void loop() {

//Verificamos si el peso ha cambiado

if(peso!=scale.get\_value(10)){

peso=scale.get\_value(10);

cliente.print("GET https://flightsregister.000webhostapp.com/conexion\_arduino.php?peso=");

cliente.print(peso);

cliente.println(" HTTP/1.0");

cliente.println();

}

}

**A continuación, se muestran los principales códigos utilizados para el desarrollo de la página web:**

**“conexion\_arduino.php”**

<?php

$db\_user="id14446876\_pstproyecto";

$db\_password="H(DAK4[(b4q\8a\*1";

$db\_name="id14446876\_refrigeradoriot";

$db\_server="localhost";

$con = mysqli\_connect($db\_server,$db\_user,$db\_password,$db\_name);

$query="INSERT INTO `Registro-Alimento` (`id`, `idSensor`, `idRefri`, `peso`, `fecha`, `hora`) VALUES (NULL, '1', '1', '".$\_GET['peso']."', CURRENT\_DATE(), CURRENT\_TIME())";

mysqli\_query($con, $query);

?>

**“loguear-php”**

<?php

session\_start();

require 'conexion.php';

$usuario = $\_POST['usuario'];

$contraseña = $\_POST['contraseña'];

$mensaje = "";

if (!empty($usuario) && !empty($contraseña)) {

$q = "SELECT \* FROM Usuario WHERE usuario = '$usuario' and contrasenia='$contraseña'";

$consulta = mysqli\_query($conexion,$q);

$filas = mysqli\_num\_rows($consulta);

$columnas = mysqli\_num\_fields($consulta);

$resultado = "";

if($filas > 0){

while($row = mysqli\_fetch\_array($consulta)){

for($i=0;$i<$columnas;$i++){

$resultado .= $row[$i];

if($i<4){

$resultado .= "--";

}

}

$resultado .= "\n";

}

}

if($filas > 0){

list($user\_id, $nom, $pass, $correo, $user) = explode("--", $resultado);

$\_SESSION['user\_id'] = $user\_id;

$\_SESSION['nombre'] = $nom;

$\_SESSION['contraseña'] = $pass;

$\_SESSION['correo'] = $correo;

$\_SESSION['user'] = $user;

header("Location: /datos-usuario.php");

}else{

$mensaje="Usuario o clave incorrecta";

header("Location: /iniciar-sesion.php");

}

}else{

$mensaje = "Por favor, llene todos los campos";

}

**“cerrar\_sesion.php”**

<?php

session\_start();

session\_destroy();

header("location: index.php");

exit();

?>

?>

**“index.php”**

<?php

session\_start();

if (isset($\_SESSION['user'])) {

header('Location: /datos-usuario.php');

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- SEO Meta Tags -->

<meta name="description" content="Create a stylish landing page for your business startup and get leads for the offered services with this HTML landing page template.">

<meta name="author" content="Inovatik">

<meta property="og:site\_name" content="" /> <!-- nombre del sitio web -->

<meta property="og:site" content="" /> <!-- link del sitio web-->

<meta property="og:title" content=""/> <!-- titulo a mostrar -->

<meta property="og:description" content="" /> <!-- descripción a mostrar -->

<meta property="og:image" content="" /> <!-- imegen para sitio web -->

<meta property="og:url" content="" />

<meta property="og:type" content="article" />

<!-- Título del sitio web -->

<title>Mi Despensa </title>

<!-- Estilos -->

<link href="https://fonts.googleapis.com/css?family=Raleway:400,400i,600,700,700i&amp;subset=latin-ext" rel="stylesheet">

<link href="css/bootstrap.css" rel="stylesheet">

<link href="css/fontawesome-all.css" rel="stylesheet">

<link href="css/swiper.css" rel="stylesheet">

<link href="css/magnific-popup.css" rel="stylesheet">

<link href="css/styles.css" rel="stylesheet">

<!--Ícono principal -->

<link rel="icon" href="images/faviconn.png">

</head>

<body data-spy="scroll" data-target=".fixed-top">

<!-- cargando -->

<div class="spinner-wrapper">

<div class="spinner">

<div class="bounce1"></div>

<div class="bounce2"></div>

<div class="bounce3"></div>

</div>

</div>

<!-- fin de código para cargar -->

<!-- navegación -->

<nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">

<!-- logo -->

<a class="navbar-brand logo-image" href="index.php"><img src="images/logoff.svg" weidth=300 ></a>

<!-- texto que acompaña al logo -->

<a class="navbar-brand logo-text page-scroll" href="index.php">Mi Despensa</a>

<!-- menú -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault" aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-awesome fas fa-bars"></span>

<span class="navbar-toggler-awesome fas fa-times"></span>

</button>

<!-- fin de menú -->

<div class="collapse navbar-collapse" id="navbarsExampleDefault">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link page-scroll" href="#header">Inicio <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link page-scroll" href="#services">Información </a>

</li>

<li class="nav-item">

<a class="nav-link page-scroll" href="#manual"> Manual </a>

</li>

<!-- menú desplegable -->

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle page-scroll" href="header" id="navbarDropdown" role="button" aria-haspopup="true" aria-expanded="false">Ingresar</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="registrarse.php"><span class="item-text">Registrarse</span></a>

<div class="dropdown-items-divide-hr"></div>

<a class="dropdown-item" href="iniciar-sesion.php"><span class="item-text">Iniciar sesión</span></a>

</div>

</li>

<!-- fin de menú desplegable -->

<li class="nav-item">

<a class="nav-link page-scroll" href="#video"> Video </a>

</li>

<li class="nav-item">

<a class="nav-link page-scroll" href="#about"> Equipo</a>

</li>

</ul>

<span class="nav-item social-icons">

<span class="fa-stack">

<a href="#your-link">

<i class="fas fa-circle fa-stack-2x facebook"></i>

<i class="fab fa-facebook-f fa-stack-1x"></i>

</a>

</span>

<span class="fa-stack">

<a href="#your-link">

<i class="fas fa-circle fa-stack-2x twitter"></i>

<i class="fab fa-twitter fa-stack-1x"></i>

</a>

</span>

</span>

</div>

</nav> <!-- fin de navegación -->

<!-- fin de navegación -->

<!-- encabezado -->

<header id="header" class="header">

<div class="header-content">

<div class="container">

<div class="row">

<div class="col-lg-6">

<div class="text-container">

<h1><span class="turquoise"> Más organización </span> Menos desperdicio </h1>

<p class="p-large">Mi Despensa le permite llevar un control de sus alimentos en el refrigerador, la cantidad que dispone y alertarlo mediante correo electrónico cuando ya no cuente con guarniciones. </p>

<a class="btn-solid-lg page-scroll" href="#services">CONOCER MÁS</a>

</div> <!-- fin de contenedor de texto -->

</div> <!-- fin de columna -->

<div class="col-lg-6">

<div class="image-container">

<img class="img-fluid" src="images/header-teamwork.svg" alt="alternative">

</div> <!-- fin de contenedor de imagen -->

</div> <!-- fin de columna -->

</div> <!-- fin del row -->

</div> <!-- fin de contenedor -->

</div> <!-- fin de encabezado -->

</header> <!-- fin de header -->

<!-- fin de header -->

<!-- Servicios -->

<div id="services" class="cards-1">

<div class="container">

<div class="row">

<div class="col-lg-12">

<h2>Mi Despensa</h2>

<p class="p-heading p-large"> Mi Despensa le permite tener un mayor control sobre los alimentos que almacena en su refrigerador</p>

</div> <!-- fin de columna -->

</div> <!-- fin de row -->

<div class="row">

<div class="col-lg-12">

<!-- Carta -->

<div class="card">

<img class="card-image" src="images/services-icon-11.svg" alt="alternative">

<div class="card-body">

<h4 class="card-title">Datos estadísticos</h4>

<p>Obtenga datos estadísticos de sus guarniciones, conozca porcentualmente la cantidad de alimentos con los que cuenta (frutas, vegetales, carnes, lacteos, etc)</p>

</div>

</div>

<!-- end of carta -->

<!-- Carta -->

<div class="card">

<img class="card-image" src="images/services-icon-22.svg" alt="alternative">

<div class="card-body">

<h4 class="card-title">Compras más ágiles</h4>

<p>Revise desde cualquier lugar la cantidad de alimentos con la que cuenta, ideal para tener siempre presente qué es lo que falta y lo que ya se tiene al momento de realizar compra de insumos.</p>

</div>

</div>

<!-- fin de carta-->

<!-- Carta -->

<div class="card">

<img class="card-image" src="images/services-icon-33.svg" alt="alternative">

<div class="card-body">

<h4 class="card-title">Avisos por correo</h4>

<p>Mantengase siempre informado del contenido de alimentos en su refrigerador, reciba alertas por correo electrónico cuando falten guarniciones o estè corto de insumos</p>

</div>

</div>

<!-- fin de carta -->

</div> <!-- fin de col -->

</div> <!-- fin de row -->

</div> <!-- fin de container -->

</div> <!-- fin de cards-1 -->

<!-- fin de services -->

<!-- Detalle 1 -->

<div id= "manual" class="basic-1">

<div class="container">

<div class="row">

<div class="col-lg-6">

<div class="text-container">

<h2>Manual de Usuario</h2>

<p>Revise el siguiente macual para conocer todas las funciones que brinda Mi Despensa </p>

<a class="btn-solid-reg popup-with-move-anim" href="#details-lightbox-1">MANUAL</a>

</div> <!-- end of text-container -->

</div> <!-- end of col -->

<div class="col-lg-6">

<div class="image-container">

<img class="img-fluid" src="images/details-1-office-worker.svg" alt="alternative">

</div> <!-- end of image-container -->

</div> <!-- end of col -->

</div> <!-- end of row -->

</div> <!-- end of container -->

</div> <!-- end of basic-1 -->

<!-- end of details 1 -->

<!-- Details Lightboxes -->

<!-- Details Lightbox 1 -->

<div id="details-lightbox-1" class="lightbox-basic zoom-anim-dialog mfp-hide">

<div class="container">

<div class="row">

<button title="Close (Esc)" type="button" class="mfp-close x-button">×</button>

<div class="col-lg-6">

<div class="image-container">

<img class="img-fluid" src="images/b9cde93e698d514d8030804e4ba0d2d8-0.jpg" alt="alternative">

</div> <!-- end of image-container -->

</div> <!-- end of col -->

<div class="col-lg-6">

<div class="image-container">

<img class="img-fluid" src="images/b9cde93e698d514d8030804e4ba0d2d8-1.jpg" alt="alternative">

</div> <!-- end of image-container -->

</div> <!-- end of col -->

</div> <!-- end of row -->

</div> <!-- end of container -->

</div> <!-- end of lightbox-basic -->

<!-- end of details lightbox 1 -->

<!-- Video -->

<div id= "video" class="basic-3">

<div class="container">

<div class="row">

<div class="col-lg-12">

<h2>Video demostrativo</h2>

</div> <!-- end of col -->

</div> <!-- end of row -->

<div class="row">

<div class="col-lg-12">

<!-- Video Preview -->

<div class="image-container">

<div class="video-wrapper">

<a class="popup-youtube" href="https://www.youtube.com" data-effect="fadeIn">

<img class="img-fluid" src="images/video-frame.svg" alt="alternative">

<span class="video-play-button">

<span></span>

</span>

</a>

</div> <!-- end of video-wrapper -->

</div> <!-- end of image-container -->

<!-- end of video preview -->

<p>Conozca las facilidades que le brinda <strong>Mi despensa</strong> y cómo hacer uso de todas las funciones que brinda este innovador proyecto</p>

</div> <!-- end of col -->

</div> <!-- end of row -->

</div> <!-- end of container -->

</div> <!-- end of basic-3 -->

<!-- end of video -->

<!-- About -->

<div id="about" class="basic-4">

<div class="container">

<div class="row">

<div class="col-lg-12">

<h2>Acerca de nuestro equipo</h2>

<p class="p-heading p-large">Grupo 1 - Programación de Sistemas Telemáticos</p>

</div> <!-- end of col -->

</div> <!-- end of row -->

<div class="row">

<div class="col-lg-12">

<!-- Team Member -->

<div class="team-member">

<div class="image-wrapper">

<img class="img-fluid" src="images/team-member-3.svg" alt="alternative">

</div> <!-- end of image-wrapper -->

<p class="p-large"><strong>Elvia Daniela De La Cruz Intriago</strong></p>

<p class="job-title">Estudiante Ing. Mecatrónica</p>

</div> <!-- end of team-member -->

<!-- end of team member -->

<!-- Team Member -->

<div class="team-member">

<div class="image-wrapper">

<img class="img-fluid" src="images/team-member-2.svg" alt="alternative">

</div> <!-- end of image-wrapper -->

<p class="p-large"><strong>Carlos Axxel Naranjo Riofrío</strong></p>

<p class="job-title">Estudiante Ing. Mecatrónica</p>

</div> <!-- end of team-member -->

<!-- end of team member -->

<!-- Team Member -->

<div class="team-member">

<div class="image-wrapper">

<img class="img-fluid" src="images/team-member-2.svg" alt="alternative">

</div> <!-- end of image-wrapper -->

<p class="p-large"><strong>Sergio Kevin Véliz Aldáz</strong></p>

<p class="job-title">Estudiante Ing. Mecatrónica</p>

</div> <!-- end of team-member -->

<!-- end of team member -->

<!-- Team Member -->

<div class="team-member">

<div class="image-wrapper">

<img class="img-fluid" src="images/team-member-2.svg" alt="alternative">

</div> <!-- end of image-wrapper -->

<p class="p-large"><strong>Frank Andrés Vera Vera</strong></p>

<p class="job-title">Estudiante Ing. Mecatrónica</p>

</div> <!-- end of team-member -->

<!-- end of team member -->

</div> <!-- end of col -->

</div> <!-- end of row -->

</div> <!-- end of container -->

</div> <!-- end of basic-4 -->

<!-- end of about -->

<!-- Footer -->

<div class="footer">

<div class="container">

<div class="row">

<div class="col-md-4">

<div class="footer-col">

<h4>Acerca de Mi Despensa</h4>

<p>Este es un proyecto de la materia Programación de Sistemas Telemáticos</p>

</div>

</div> <!-- end of col -->

<div class="col-md-4">

<div class="footer-col middle">

<h4>Integrantes</h4>

<ul class="list-unstyled li-space-lg">

<li class="media">

<i class="fas fa-square"></i>

<div class="media-body">Elvia De La Cruz <a class="turquoise" href="#your-link"></a></div>

</li>

<li class="media">

<i class="fas fa-square"></i>

<div class="media-body">Carlos Naranjo <a class="turquoise" href="#your-link"></a></div>

</li>

<li class="media">

<i class="fas fa-square"></i>

<div class="media-body">Sergio Véliz <a class="turquoise" href="#your-link"></a></div>

</li>

<li class="media">

<i class="fas fa-square"></i>

<div class="media-body">Frank Vera <a class="turquoise" href="#your-link"></a></div>

</li>

</ul>

</div>

</div> <!-- end of col -->

<div class="col-md-4">

<div class="footer-col last">

<h4>Redes Sociales</h4>

<span class="fa-stack">

<a href="#your-link">

<i class="fas fa-circle fa-stack-2x"></i>

<i class="fab fa-facebook-f fa-stack-1x"></i>

</a>

</span>

<span class="fa-stack">

<a href="#your-link">

<i class="fas fa-circle fa-stack-2x"></i>

<i class="fab fa-twitter fa-stack-1x"></i>

</a>

</span>

</div>

</div> <!-- end of col -->

</div> <!-- end of row -->

</div> <!-- end of container -->

</div> <!-- end of footer -->

<!-- end of footer -->

<!-- Copyright -->

<div class="copyright">

<div class="container">

<div class="row">

<div class="col-lg-12">

<p class="p-small">Copyright © 2020 <a href="https://inovatik.com">Inovatik</a> - All rights reserved</p>

</div> <!-- end of col -->

</div> <!-- enf of row -->

</div> <!-- end of container -->

</div> <!-- end of copyright -->

<!-- end of copyright -->

<!-- Scripts -->

<script src="js/jquery.min.js"></script> <!-- jQuery for Bootstrap's JavaScript plugins -->

<script src="js/popper.min.js"></script> <!-- Popper tooltip library for Bootstrap -->

<script src="js/bootstrap.min.js"></script> <!-- Bootstrap framework -->

<script src="js/jquery.easing.min.js"></script> <!-- jQuery Easing for smooth scrolling between anchors -->

<script src="js/swiper.min.js"></script> <!-- Swiper for image and text sliders -->

<script src="js/jquery.magnific-popup.js"></script> <!-- Magnific Popup for lightboxes -->

<script src="js/validator.min.js"></script> <!-- Validator.js - Bootstrap plugin that validates forms -->

<script src="js/scripts.js"></script> <!-- Custom scripts -->

</body>

</html>

# **Análisis de presupuesto**

A continuación, se presenta el presupuesto estimado para la implementación de la automatización del refrigerador con un contenedor de alimentos:

|  |  |
| --- | --- |
| **Material** | **Costo Unitario** |
| Contenedor plástico | 1,25 |
| Celda de carga 50 kg + controlador HX711 | 10,00 |
| Arduino UNO | 10,00 |
| Ethernet Shield W5100 | 11,00 |
| **TOTAL** | **32,25** |

El precio inicial para la automatización obtenido fue de 32,25, lo cuál incluye la instalación de un solo contenedor. La instalación de contenedores adicionales tendría un valor estimado de 11,25 por contenedor (esto representa el valor de la estructura plástica y la celda de carga requerida).

# **Conclusiones**

* El diseño de solución planteado facilitará el control de despensa en el refrigerador para el usuario, de manera que las cantidades de alimentos disponibles cambien dinámicamente en una base de datos sin la necesidad de que el usuario ingrese manualmente cambio a cambio dicha información, ahorrándole tiempo y trabajo brindándole información útil y accesible desde cualquier lugar.
* Se logró plantear una nueva alternativa de aplicación tecnológica en el proceso de control de guarniciones en refrigerador, basada en el uso de celdas de carga en lugar de cámaras fotográficas, que es lo que se suele ver en aplicaciones de este tipo para el control de alimentos en refrigeradores.
* El presente proyecto demuestra la importancia de la combinación de hardware y aplicativos webs para facilitar mediante la automatización las actividades cotidianas de las personas.

# **Referencias bibliográficas**

* Casa Domótica. (11 de Noviembre de 2016). *Casa digital*. Obtenido de La comodidd de la domótica: https://www.casasdigitales.com/neveras-inteligentes/
* Llamas, L. (6 de Febrero de 2017). *Ingeniería, Informática y Diseño*. Obtenido de CONECTAR ARDUINO A INTERNET O LAN CON SHIELD ETHERNET W5100: https://www.luisllamas.es/arduino-ethernet-shield-w5100/
* Pereira, M. (14 de Marzo de 2020). *Amigastronómicas*. Obtenido de ALACENA, LA APP QUE TE ORGANIZA LA DESPENSA: http://www.amigastronomicas.com/2016/04/04/alacena-la-app-te-organiza-la-despensa/
* San Martín, E. (10 de Enero de 2019). *Consumer*. Obtenido de Inventario de comida en casa: https://www.consumer.es/economia-domestica/sociedad-consumo/como-hacer-un-inventario-de-comida-en-casa.html
* The Food and Agriculture Organization of the United Nations. (15 de Noviembre de 2013). *Food Wastage footprint.* Obtenido de Food wastage footprint: Impacts on natural resources - Summary report: http://www.fao.org/3/i3347e/i3347e.pdf

1. The Food and Agriculture Organization of the United Nations. (15 de Noviembre de 2013). Food Wastage footprint. Obtenido de Food wastage footprint: Impacts on natural resources - Summary report: http://www.fao.org/3/i3347e/i3347e.pdf [↑](#footnote-ref-1)