



**Nombre:**

Carlos Ramírez (2018-6055)

**Materia:**

Programación III

**Tema:**

Proyecto #3

Aplicación Web de Reconocimiento Facial  
Búsqueda de Personas Desaparecidas.

Entregable #2:

Diagramas de la Arquitectura  
Análisis de Tecnologías

**Profesor(a):**

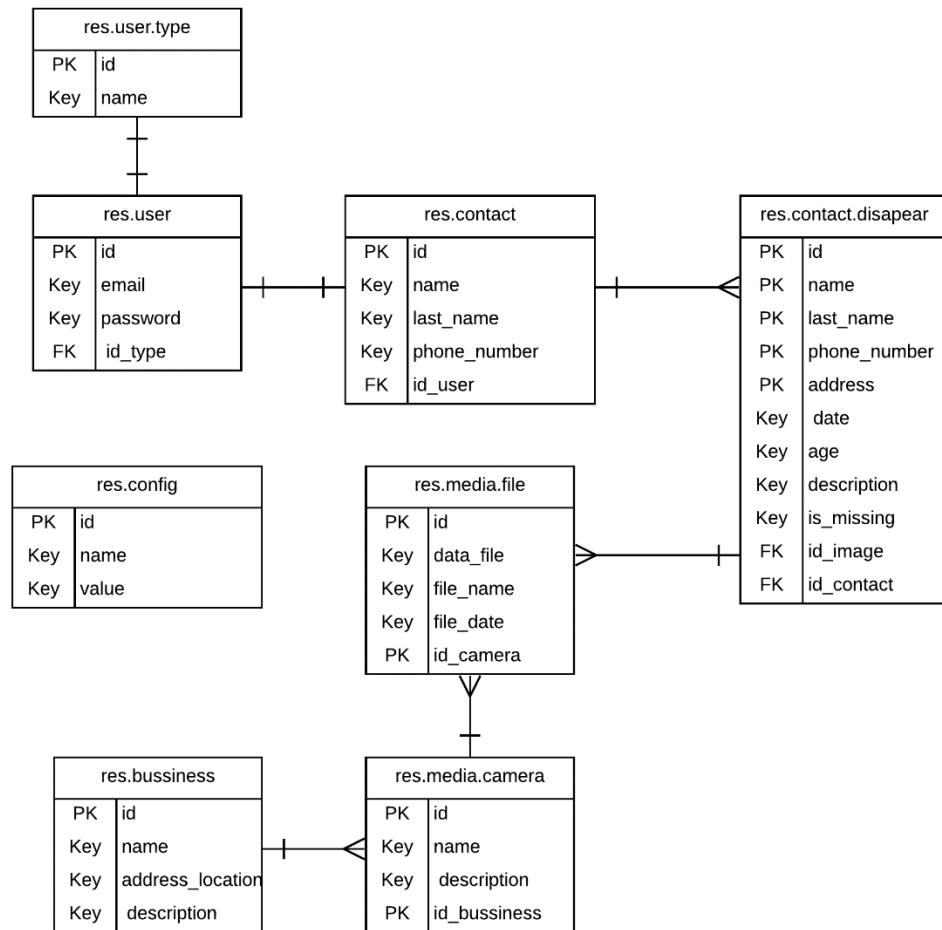
Willis Polanco

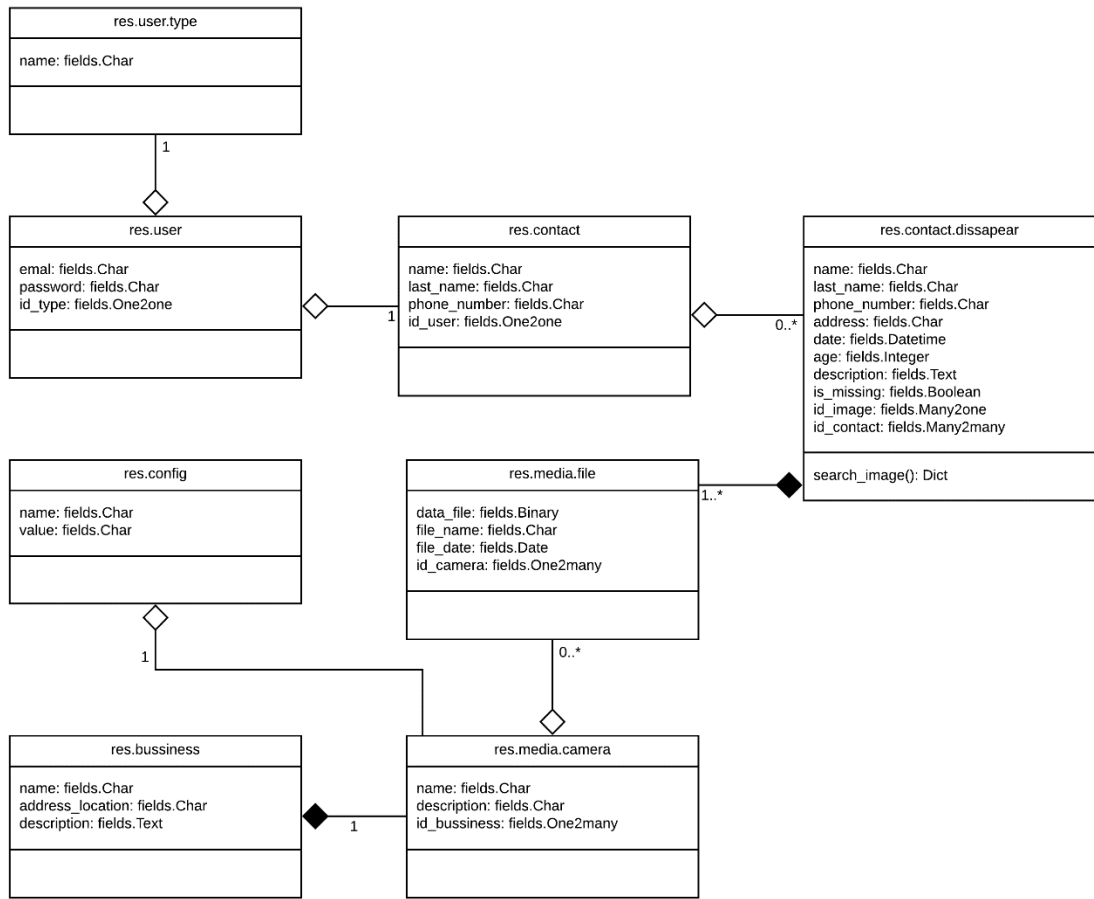
*La Caleta, Boca Chica, Santo Domingo de Guzmán*

*6 de marzo del 2020*

*Cuatrimestre 2020-C-1*

## Diagrama diseño arquitectura de la aplicación.





## Informe sobre tecnologías a utilizar.

### Propuesta tecnológica 1.

#### **Tecnologías:**

- a) C#
- b) Asp.net
- c) MVC
- d) Entity Framework
- e) Bootstrap
- f) Microsoft Sql Server
- g) Visual Studio
- h) Microsoft Sql Server Managment Studio
- i) Python
- j) OpenCV
- k) Microsoft Azure
- l) Deep Learning

#### **Approach:**

Se utilizará C# junto con Asp.net valiéndose del patrón de diseño MVC y Entity Framework para crear la estructura server-side de la aplicación web. Usando Bootstrap como framework de CSS para estilizar la página.

Para la lógica de reconocimiento facial y búsqueda de personas se creará una API en Python la cual recibirá la imagen y luego de realizar la lógica necesaria retornara un diccionario con la dirección de la cámara en donde se encontró la persona en la imagen, fecha y la imagen del momento donde se identificó. Así como también informacion técnica del macheo como porcentaje de parecido etc. todo esto en el Http response.

Para el mecanismo de reconocimiento se utilizará la librería OpenCV y Deep Learning de manera que la aplicación pueda mejorar los mecanismos de reconocimiento.

#### **Ventajas:**

1. Generación automática de los modelos de manera automática a partir de la base de datos.
2. Generación automática de los métodos típicos (create, update, select, delete).
3. Generación automática de las views con sus respectivos controladores.
4. Gran integración base de datos/framework.
5. Gran cantidad de recursos de estilización.
6. Gran Cantidad de documentación y tutoriales.

## **Desventajas:**

1. La lógica de negocios se debe implementar manualmente.
2. No dispone de lógica por default para la subida de archivos.
3. Diseño de vistas muy limitados.
4. Se debe personalizar las vistas que vienen por default.
5. Se debe crear y codificar la lógica para el login y el Signup
6. Se necesita hostear el API y conectar con ella en cada búsqueda.
7. Posibles contratiempos de integración entre Python y C#
8. Necesidad de Conocer los dos lenguajes.

## Propuesta tecnológica 2.

### **Tecnologías**

- a) Python
- b) Odoo ERP
- c) OpenCV
- d) Deep Learning
- e) PostgreSQL
- f) Q-Web

### **Approach**

Se utilizará Python como lenguaje de programación y el Enterprise Resource Manager Odoo para escribir la aplicación sobre el mismo. Lo que incluye lógica de negocios, estructura tanto de la base de datos como de los modelos, vistas, controladores, endpoints, gateways, etc.

### **Ventajas**

1. Uso de un solo lenguaje de programación.
2. Gran Cantidad de documentación y tutoriales.
3. Eliminación de los posibles errores al momento de integrar OpenCV en un API.
4. Reducción significativa del tiempo de desarrollo.
5. Creación automática de toda la estructura de la base de datos.
6. Creación automática de todos los widgets, vistas, menus, links necesarios para mostrar los modelos definidos.
7. Eliminación de la necesidad de programar y desarrollar el login, sign up etc.
8. Reutilización de modelos ya definidos como el de usuarios, contactos.
9. Odoo maneja el hosting de la página.
10. Eliminación de contratiempos de hosting.
11. Gran facilidad para definir vistas simplemente utilizando el motor de plantillas de Odoo.
12. Frontend completo de caja, no es necesario codificar el client-side.
13. Manejo de fabrica de archivos, imágenes, etc.
14. Gran escalabilidad valiéndose de su naturaleza modular.

15. Capacidades para el envío de mensajes tanto interna como externamente de fabrica sin necesidad de programacion simplemente haciendo herencia del modelo.

Propuesta seleccionada a utilizar.

<< Propuesta tecnológica 2. >>>

## Resources

<https://www.odoo.com/documentation/12.0/setup/install.html>

<https://www.odoo.com/documentation/12.0/howtos/backend.html>

<https://www.odoo.com/documentation/12.0/reference/reports.html>

<https://www.odoo.com/documentation/12.0/reference/orm.html>

<https://hackernoon.com/face-recognition-using-opencv-a-step-by-step-guide-to-build-a-facial-recognition-system-8da97cd89847>

[https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html)

<https://www.youtube.com/channel/UCfzLCWGWYyIQ0aLC5w48gBQ>