

Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree in Computer Engineering)

Curso: Proyecto de Diseño en Ingeniería en Computadores – CE5302



Documento de Diseño

(Design document)

Realizado por:

Made by:

Oscar Josué Ulate Alpízar, 201229559

Profesor:

(Professor)

Gustavo Adolfo Cubas Euceda

Fecha: 15 de marzo del 2019

(Date: March 15, 2019)

Historial de cambios

Fecha	Versión	Descripción	Autor(es)
27/02/2019	1.0	Inicio de secciones 1 y 2	Oscar Ulate
28/02/2019	1.2	Secciones 3 y 5	Oscar Ulate
16/03/2019	1.3	Primera versión entregable	Oscar Ulate

Tabla de contenido

Historial de cambios	2
1. Introducción	5
1.1. Propósito	5
1.2. Alcance	5
1.3. Contexto	5
1.4. Resumen	5
2. Referencias.....	5
3. Glosario / Acrónimos.....	6
4. Partes identificadas y sus consideraciones.....	6
5. Vista de composición.....	7
5.1. Consideraciones de diseño	7
5.2. Elementos de diseño	7
Módulo de cámaras:	7
Módulo de estampado de tiempo:	7
Módulo de buffer de memoria:.....	8
Módulo de comunicación:.....	8
Módulo de integración:.....	8
Hardware:	8
6. Lógica.....	9
6.1. Elementos de diseño	9
StartCapture (Clase):	9
ConfigureCapture (Clase):	9
CaptureImages (Clase):	9
ProcessImages (Clase):	9
Communication (Clase):	9

Recieve_Images:.....	9
Send_Images:.....	10
OpenCV:	10
7. Dependencias.....	10
8. Información.....	10
9. Uso de patrones	10
10. Vista de interfaces.....	10
11. Estructura	11
12. Interacción	11
13. Dinámica de estados	11
14. Algoritmos	11
14.1. Consideraciones de diseño	11
14.2. Elementos de diseño	11
Crear estampado de tiempo e identificador de cámara	12
15. Recursos.....	12

1. Introducción

1.1. Propósito

Los detalles de diseño presentados en este documento se enfocan en el diseño de un producto capaz de tomar fotogramas de manera síncrona entre tres cámaras FLIR Blackfly[®]S y su envío con estampado de tiempo a una computadora central. Para lograr esto se presenta descripción de los elementos involucrados desde diferentes vistas.

1.2. Alcance

Al mostrar el producto desde distintos puntos de vista, se busca materializar los requerimientos presentados en el documento de requerimientos. Cada vista aporta la información necesaria para que los desarrolladores de la aplicación puedan seguir directrices deseadas por las partes interesadas.

1.3. Contexto

Lograr presentar los detalles técnicos que conlleva diseñar cada requerimiento es clave para lograr completar las metas planteadas. Al lograr detallar el diseño desde distintos puntos de vista de diseño, es posible implementar una solución para lograr la sincronización y envío de datos de las cámaras.

1.4. Resumen

Como parte del sistema de reastreo de mariposas en tiempo real, es importante que todas las fotografías sean con una diferencia de tiempo lo más pequeña posible permitida por la tecnología de las cámaras. Este bloque debe ser transparente a los usuarios finales, investigadores y estudiantes de CICANUM de la UCR, que desean que esto funcione de manera sencilla, conectando el cable del disparador a todas las cámaras. Otros interesados, somos el equipo del SIP-Lab, donde se desea que se apliquen conocimientos en diversas áreas: matemáticas, visión por computadora, sistemas embebidos y programación.

2. Referencias

[1] Laboratorio de Procesamiento de Imágenes y Señales,
<http://www.ie.tec.ac.cr/palvarado/pmwiki/index.php/Main/HomePage>.

[2] Entrelazamiento cuántico entre el campo magnético terrestre y la migración de las mariposas Monarcas, <https://vinv.ucr.ac.cr/sigpro/web/projects/B6220>.

[3] OpenCV Library, <https://opencv.org/>.

3. Glosario / Acrónimos

CICANUM: Centro de Investigación en Ciencias Atómicas, Nucleares y Moleculares.

SIP-Lab: Signals and Images Processing Laboratory

UCR: Universidad de Costa Rica

TEC: Tecnológico de Costa Rica

4. Partes identificadas y sus consideraciones

Miembros del CICANUM: Utilizar la aplicación desarrollada, que de manera transparente al usuario, utiliza un sincronizador de cámaras para que las fotografías se tomen al mismo tiempo. Además, toda la información debe llegar a la computadora central de manera transparente para ellos.

Miembros del SIP-Lab: El principal objetivo del laboratorio es crear conocimiento en las áreas de conocimiento antes mencionadas. Para que el proyecto sea exitoso para el laboratorio, se debe crear toda la ingeniería con conocimientos investigados por los estudiantes. Además se debe crear una documentación que permita el futuro estudio, y mantenimiento de la ingeniería. Todos los requerimientos planteados por el laboratorio se encuentran en el documento de Requerimientos.

5. Vista de composición

5.1. Consideraciones de diseño

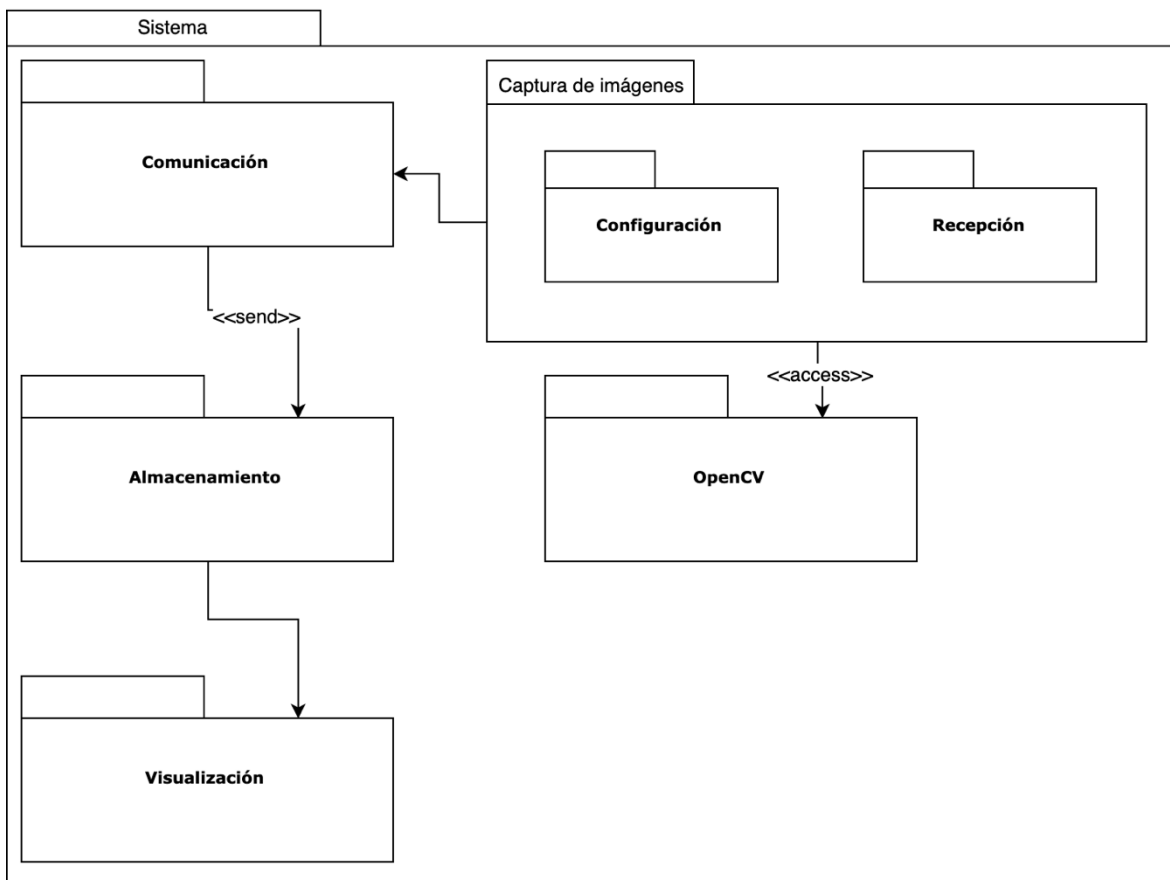


Figura 1. Bloques del diseño.

5.2. Elementos de diseño

Módulo de cámaras:

Módulo encargado de recibir los fotogramas por parte de las cámaras a una velocidad regulada por el usuario del sistema.

Módulo de estampado de tiempo:

Módulo encargado de asignarle a cada imagen una estampa del tiempo en que fue recibida. Esto para que cuando las imágenes lleguen a la aplicación en la computadora central, se puedan agrupar con las otras dos imágenes de las otras dos cámaras y aplicar sobre ellas el algoritmo de rastreo de las mariposas.

Módulo de buffer de memoria:

Este módulo es el encargado de almacenar en memoria todas las imágenes que son capturadas para luego ser enviadas a la computadora central. Este módulo debe ser replicado en todas las tarjetas de desarrollo que tengan conectadas una cámara.

Módulo de comunicación:

Este módulo es el encargado de enviar las fotografías a la computadora central. Se debe investigar la mejor manera de hacerlo, pero por el momento se considera enviarlas por medio del protocolo TCP/IP.

Módulo de integración:

Este módulo integra todos los anteriores con el sistema ya existente, creado en una iteración pasada de este proyecto. Debe permitir al usuario final utilizar todos los módulos anteriores de una manera transparente.

Hardware:

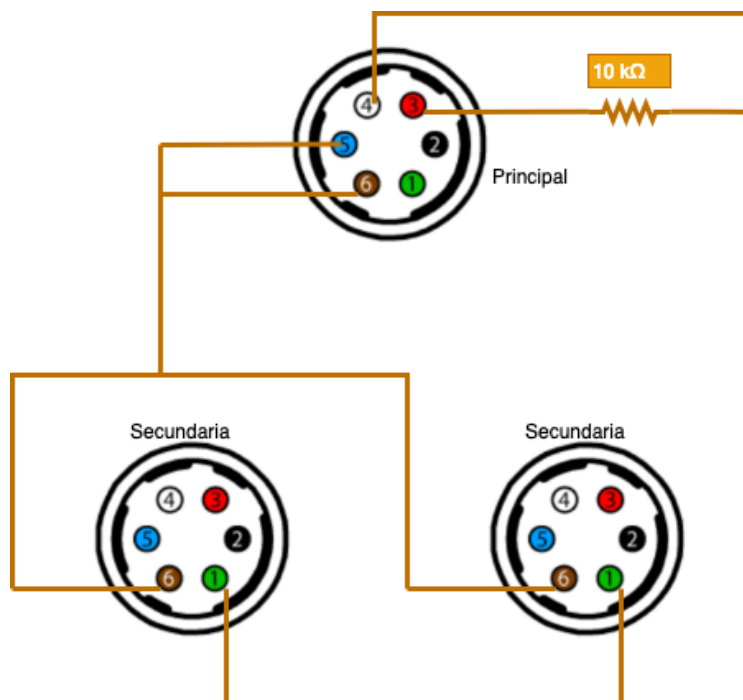


Figura 2. Diagrama del disparador de captura de imágenes.

Este módulo, mostrado en la figura anterior, es el encargado de sincronizar la captura de las imágenes sin la necesidad de usar software. El *Módulo de cámaras* es el encargado de recibir las imágenes capturadas cuando se activa estos disparadores.

6. Lógica

Se utiliza un diagrama de clases UML, con perspectiva de especificación, para mostrar la estructura lógica del sistema y las relaciones a un nivel más cercano a la implementación. Durante la implementación se puede generar el mismo diagrama de clases, pero a un nivel de implementación.

6.1. Elementos de diseño

StartCapture (Clase):

Esta clase se encuentra en el *AppCore* creado en una iteración pasada de este sistema. Esta es la encargada de comenzar la captura de las imágenes en todas las cámaras de manera simultánea con la configuración brindada por el usuario por medio de la interfaz gráfica.

ConfigureCapture (Clase):

Clase que permite al usuario configurar la captura de imágenes, así como almacenar los fotogramas en el disco duro de la computadora principal si el usuario lo desea.

CaptureImages (Clase):

Esta clase debe estar en todas las tarjetas de desarrollo que tengan una cámara conectada a ellas. Su función es iniciar la captura de imágenes y recibirlas apropiadamente, para posteriormente procesarles los algoritmos necesarios.

ProcessImages (Clase):

Esta clase tiene como función aplicarle todos los algoritmos a las imágenes e incluirlas en el buffer de imágenes para posteriormente ser enviadas a la computadora principal en la siguiente clase. Por el momento, el principal algoritmo que se aplica en esta clase es el de aplicarle a cada imagen su estampado de tiempo.

Communication (Clase):

Esta clase se debe diseñar en dos partes, la recepción de las imágenes y el envío de estas. La razón por la que no se separa en módulos distintos para cada tarjeta de desarrollo es porque se desea implementar el diseño tal que cualquier computadora sea la principal. Esto facilitaría la configuración para el usuario final.

Recieve_Images:

El método *Recieve_Images* se encarga de recibir las imágenes por medio del protocolo seleccionado. El diseño original propone usar TCP/IP, pero podría cambiarse.

Send_Images:

El método Send_Images se encarga de enviar las imágenes por medio del protocolo seleccionado. El diseño original propone usar TCP/IP, pero podría cambiarse.

OpenCV:

Esta es una biblioteca que simplifica la manipulación de imágenes. Esto va a permitir incluir el estampado de tiempo, o otros métodos necesarios para el procesamiento de las mismas.

7. Dependencias

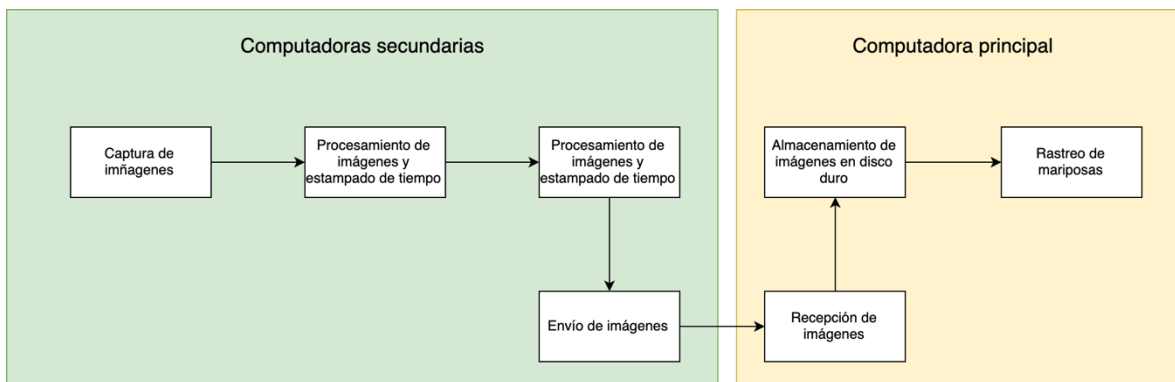


Figura 3. Dependencia de módulos.

En la Figura 3 se muestra el flujo de datos que debe seguir esta parte del proceso del Proyecto para que sea exitoso. Todos los pasos deben ser ejecutados siempre que se desee capturar nuevamente el vuelo de las mariposas. El último paso, *Rastreo de mariposas*, que se está desarrollando en paralelo a este proyecto, puede ejecutarse sin la necesidad de ejecutar todos los pasos anteriores, puesto que se puede basar en tomas de imagen realizadas previamente y almacenadas en el disco duro.

8. Información

No aplica.

9. Uso de patrones

No aplica.

10. Vista de interfaces

No aplica.

11.Estructura

No aplica.

12.Interacción

No aplica.

13.Dinámica de estados

No aplica.

14.Algoritmos

14.1. Consideraciones de diseño

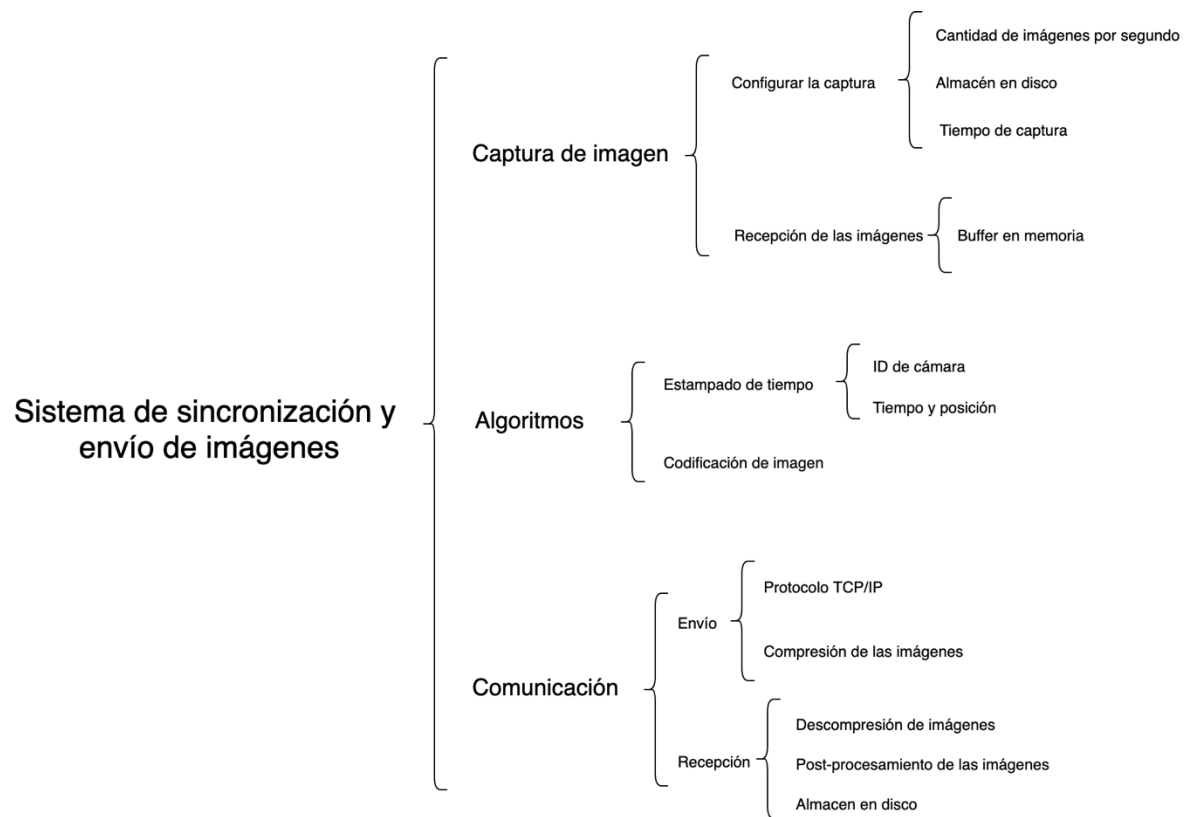


Figura 4. Diagrama de Warnier.

14.2. Elementos de diseño

En la Figura 4, se muestra los posibles escenarios que puede existir en el sistema.

Crear estampado de tiempo e identificador de cámara

Crear el estampado de tiempo, junto con el identificador de la cámara es esencial para el posterior proceso en el sistema de reconocer el vuelo de las mariposas. Sin esta información en cada imagen, es imposible que el proceso de identificación se genere de manera correcta. OpenCV [3], biblioteca especializada en el manejo de imágenes, es la indicada para agregar esos encabezados a las imágenes. Además, es necesario diseñar el módulo que comprima las imágenes para ahorrar ancho de banda en medio de comunicación establecido, que hasta el momento es TCP/IP.

Por otro lado, para el identificador de las cámaras, se va a usar el SDK propietario de FLIR para las cámaras. Con este se puede obtener el identificador de cada cámara para que a la hora de procesar las imágenes en el siguiente paso de identificación de mariposas, se pueda acomodar las imágenes en el orden adecuado.

15. Recursos

No aplica.