

# SISTEMA DE MONITORIZACIÓN AMBIENTAL BASADO EN VIDEO: HORUS

## Director:

Andrés Fernando Osorio Arias, Ph.D.  
[<afosorioar@unal.edu.co>](mailto:<afosorioar@unal.edu.co>)

## Desarrollo:

Cristian Andrés Ortiz Alarcón, M.Ing.  
[<cristian.ortiz.alarcon@gmail.com>](mailto:<cristian.ortiz.alarcon@gmail.com>)

Juan Camilo Pérez Muñoz, M.Sc.      Sebastián Múnera Álvarez, Ing  
[<jcperezmu@gmail.com>](mailto:<jcperezmu@gmail.com>)      [<sfmunera@unal.edu.co>](mailto:<sfmunera@unal.edu.co>)

César Augusto Cartagena Ocampo, Ing  
[<cacartag@unal.edu.co>](mailto:<cacartag@unal.edu.co>)



# Contenido

<b>1. Definiciones</b>	<b>5</b>
<b>2. Introducción</b>	<b>17</b>
<b>3. Diseño de HORUS</b>	<b>22</b>
3.1. Estructura del Toolbox . . . . .	22
3.2. Base de datos . . . . .	24
3.2.1. Modelo Entidad–Relación . . . . .	24
3.2.2. Entidades . . . . .	26
<b>4. Instalación</b>	<b>35</b>
4.1. Instalación Básica . . . . .	35
4.2. Base de Datos . . . . .	35
4.2.1. Configuración mediante la interfaz gráfica <i>Setup database</i> . . . . .	35
4.2.2. Configuración manual . . . . .	36
<b>5. GUI</b>	<b>39</b>
5.1. Setup database . . . . .	40
5.2. Edit database . . . . .	40
5.2.1. Estación . . . . .	41
5.2.2. Cámara . . . . .	45
5.2.3. GCP . . . . .	45
5.2.4. Sensor . . . . .	50
5.2.5. Tipo de Medición . . . . .	51
5.2.6. Generar rutas de imágenes . . . . .	57
5.3. Create new user . . . . .	57
5.4. Configure cameras . . . . .	58
5.5. Setup image paths . . . . .	59
5.6. Configure rectification . . . . .	62
5.7. Create new ROI . . . . .	66
5.8. Manually rectify images . . . . .	69
5.9. Configure fusion . . . . .	72
5.10. Manually merge images . . . . .	77
5.11. Configure capture automatic . . . . .	80
5.12. Configure processing automatic . . . . .	85
5.13. Configure database synchronization automatic . . . . .	88
5.14. Configure thumbnail generation and upload . . . . .	93

<b>6. FAQ</b>	<b>96</b>
6.1. ¿Dónde puedo obtener más información sobre el Proyecto HORUS? . . . . .	96
6.2. ¿Cómo puedo contactar al equipo HORUS? . . . . .	96
6.3. ¿Cómo puedo acceder a la información de las estaciones monitorizadas por el Proyecto HORUS? . . . . .	96
6.4. ¿Qué requerimientos necesito para ejecutar las rutinas del Toolbox? . . . . .	96
6.5. ¿En qué sistema operativo funciona el Toolbox? . . . . .	97
6.6. ¿Dónde puedo obtener documentación de las rutinas? . . . . .	97
6.7. ¿Puedo integrar mis desarrollos al paquete HORUS? . . . . .	97
6.8. ¿Qué limitaciones tiene el uso de este toolbox? . . . . .	97
6.9. ¿Ustedes diseñan e instalan estaciones? . . . . .	97
6.10. ¿Si yo tengo una estación la puedo integrar al portal HORUS? . . . . .	97
6.11. ¿Quién diseñó el sistema? . . . . .	97
6.12. ¿En el marco de qué proyecto se desarrolló esta versión y quien lo financió? . . . . .	98

# Listado de Figuras

1.1.	Relación entre $(u, v)$ y $(x, y, z)$ . . . . .	5
1.2.	Ejemplo de un ROI en una imagen . . . . .	6
1.3.	Ejemplo de imagen oblicua . . . . .	6
1.4.	Ejemplo de imagen rectificada . . . . .	7
1.5.	Ejemplo de imagen fusionada . . . . .	7
1.6.	Ejemplo de imagen <i>snap</i> . . . . .	8
1.7.	Ejemplo de imagen <i>timex</i> . . . . .	8
1.8.	Ejemplo de imagen <i>variance</i> . . . . .	9
1.9.	Mapa de resolución de una imagen oblicua . . . . .	10
1.10.	Diagrama de flujo de la rectificación en HORUS . . . . .	11
1.11.	Esquema de la fusión entre dos imágenes . . . . .	13
1.12.	Diagrama de flujo de la fusión en HORUS . . . . .	14
1.13.	Selección de una región para un timestamp . . . . .	15
2.1.	Esquema general de HORUS . . . . .	18
2.2.	Esquema de funcionamiento completo de HORUS . . . . .	19
2.3.	Línea de tiempo de los procesos . . . . .	21
3.1.	Esquema general del sistema de archivos de HORUS . . . . .	22
3.2.	Diagrama de comunicación entre los módulos . . . . .	23
3.3.	Modelo ER de HORUS . . . . .	25
4.1.	Interfaz para configuración inicial de la base de datos . . . . .	35
5.1.	Interfaz principal de HORUS . . . . .	39
5.2.	Interfaz para ingresar los datos de usuario para el login . . . . .	40
5.3.	Interfaz para insertar una nueva estación . . . . .	41
5.4.	Interfaz para actualizar estación . . . . .	41
5.5.	Diagrama de flujo interfaz DB Editor - Estación . . . . .	43
5.6.	Interfaz para insertar una nueva cámara . . . . .	44
5.7.	Interfaz para actualizar una cámara . . . . .	44
5.8.	Diagrama de flujo interfaz DB Editor - Cámara . . . . .	46
5.9.	Interfaz para insertar un nuevo GCP . . . . .	47
5.10.	Interfaz para actualizar un GCP . . . . .	47
5.11.	Estructura de excel para importar GCPs . . . . .	48
5.12.	Diagrama de flujo interfaz DB Editor - GCP . . . . .	49
5.13.	Interfaz para insertar un nuevo sensor . . . . .	50
5.14.	Interfaz para actualizar un sensor . . . . .	50
5.15.	Diagrama de flujo interfaz DB Editor - Sensor . . . . .	52
5.16.	Interfaz para insertar un nuevo tipo de medición . . . . .	53
5.17.	Interfaz para actualizar un tipo de medición . . . . .	53
5.18.	Estructura de MATLAB si el tipo de medición es serie . . . . .	54

5.19. Estructura de Excel si el tipo de medición es matriz . . . . .	55
5.20. Diagrama de flujo interfaz DB Editor - Tipo de Medición . . . . .	56
5.21. Interfaz para generar las rutas automáticamente . . . . .	57
5.22. Interfaz para crear un nuevo usuario con permisos limitados . . . . .	57
5.23. Interfaz de Configuración de cámaras . . . . .	59
5.24. Diagrama de flujo para el proceso de configuración de las cámaras . . . . .	60
5.25. Interfaz inicial para editar las rutas . . . . .	60
5.26. Interfaz para editar las rutas con los campos rellenos . . . . .	61
5.27. Diagrama de flujo interfaz Path Editor . . . . .	62
5.28. Proceso de calibración . . . . .	63
5.29. Interfaz de Calibración . . . . .	64
5.30. Ejemplo de archivo para importar GCPs marcados . . . . .	65
5.31. Diagrama de flujo del proceso de generación de calibración . . . . .	67
5.32. Interfaz para insertar un nuevo ROI . . . . .	67
5.33. Interfaz para actualizar un ROI . . . . .	68
5.34. Diagrama de flujo interfaz ROI Tool . . . . .	70
5.35. Interfaz de Rectificación . . . . .	71
5.36. Diagrama de flujo del proceso de rectificación en lote . . . . .	73
5.37. Interfaz de Calibración de fusión . . . . .	73
5.38. Ejemplo de archivo para importar puntos comunes marcados . . . . .	75
5.39. Diagrama de flujo del proceso de generación de parámetros de fusión . . . . .	76
5.40. Interfaz de Fusión . . . . .	77
5.41. Diagrama de flujo del proceso de fusión en lote . . . . .	79
5.42. Interfaz de Configuración de capturas . . . . .	80
5.43. Diagrama de flujo del proceso de configuración de captura . . . . .	82
5.44. Diagrama de flujo del automático de captura . . . . .	84
5.45. Interfaz para configurar el automático de procesamiento . . . . .	86
5.46. Diagrama de flujo del proceso de configuración de procesamiento . . . . .	88
5.47. Diagrama de flujo del automático de procesamiento . . . . .	89
5.48. Interfaz para configurar el automático de sincronización . . . . .	90
5.49. Interfaz inicial para generar miniaturas . . . . .	92
5.50. Interfaz con los datos necesarios para generar miniaturas . . . . .	92
5.51. Interfaz para subir las miniaturas generadas a la web . . . . .	94
5.52. Diagrama de flujo interfaz Thumbnails Tool . . . . .	95

# Capítulo 1

## Definiciones

En este capítulo se introducirán los términos utilizados en todo el documento que podrían no ser muy claros para el lector. Estos términos se mencionan en numerosas ocasiones y son necesarios para entender los procesos que se llevan a cabo en el sistema HORUS.

**GCP o *Ground Control Point* por su sigla en inglés** Puntos de control georreferenciados (generalmente medidos con GPS) asociados a una estación, representados por una coordenada  $(x, y, z)$ . Los GCPs son usados principalmente para mapear sus coordenadas  $(x, y, z)$  con coordenadas  $(u, v)$  (medidas en píxeles) relativos al plano de una imagen, escogidas por el usuario. En la figura 1.1 se muestra la relación entre las coordenadas  $(x, y, z)$  de un GCP y las coordenadas  $(u, v)$  en el plano de una imagen cuyo origen está en la esquina superior izquierda. El eje  $u$  es positivo de izquierda a derecha, y el eje  $v$  es positivo hacia abajo.

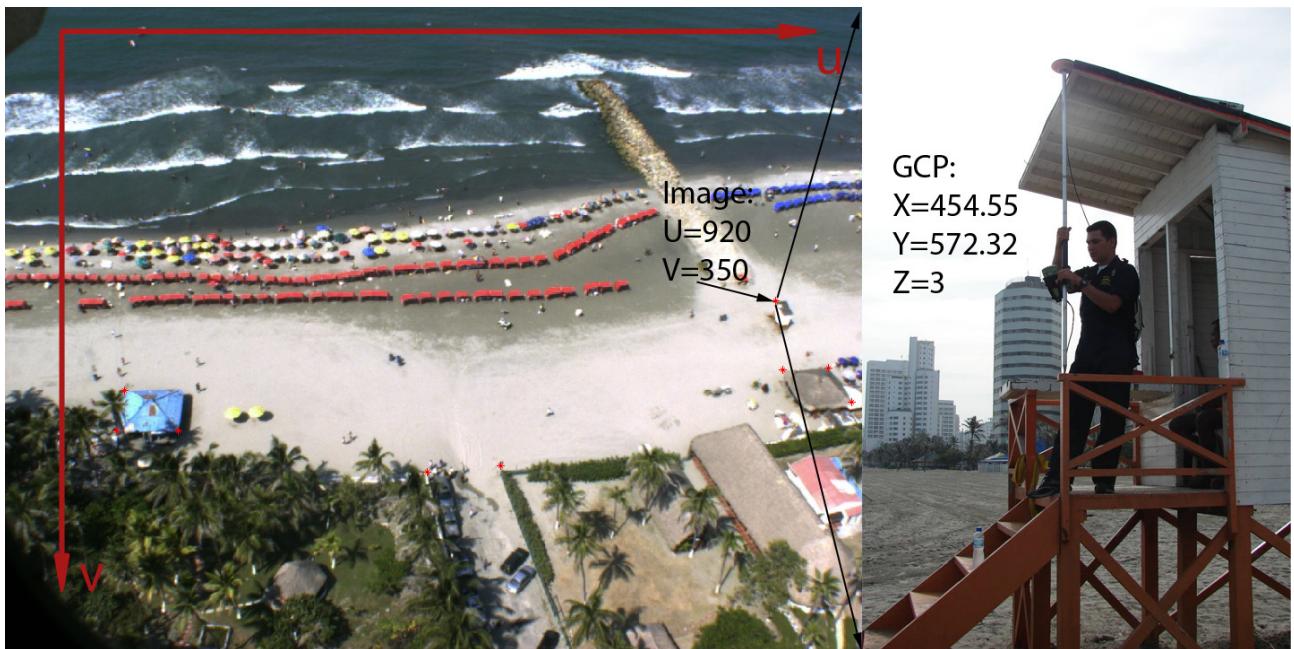


Figura 1.1: Relación entre  $(u, v)$  y  $(x, y, z)$

**ROI o *Region Of Interest* por su sigla en inglés** Un ROI es un polígono dentro de una imagen que delimita una región de la misma. El polígono es representado como un conjunto de vértices con coordenadas  $(u, v)$  (medidos en píxeles) relativos al plano de la imagen, tales que cada par de vértices consecutivos están conectados por una línea recta. En la

figura 1.2 se muestra un ejemplo de ROI en una imagen costera, donde se pueden identificar los vértices (puntos amarillos) y las líneas que los unen (líneas rojas) que conforman el polígono.



Figura 1.2: Ejemplo de un ROI en una imagen

**Calibración** Es el nombre que se le da al proceso de aplicar un método de optimización para encontrar los parámetros de rectificación y fusión. La calibración involucra la elección del ROI, GCPs, y otros parámetros necesarios para el método.

**Imagen Oblicua** Es aquella imagen capturada por una cámara con un ángulo de inclinación, de tal manera que la imagen tenga una perspectiva. En la figura 1.3 se muestra una imagen oblicua.

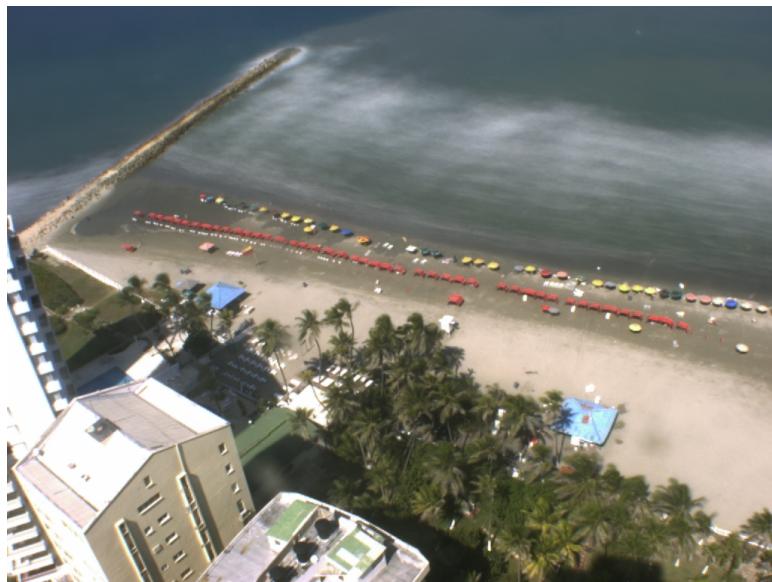


Figura 1.3: Ejemplo de imagen oblicua

**Imagen Rectificada** Es una imagen oblicua que sufre una transformación para que parezca como si hubiera sido capturada perpendicularmente sobre el plano de la zona de estudio,

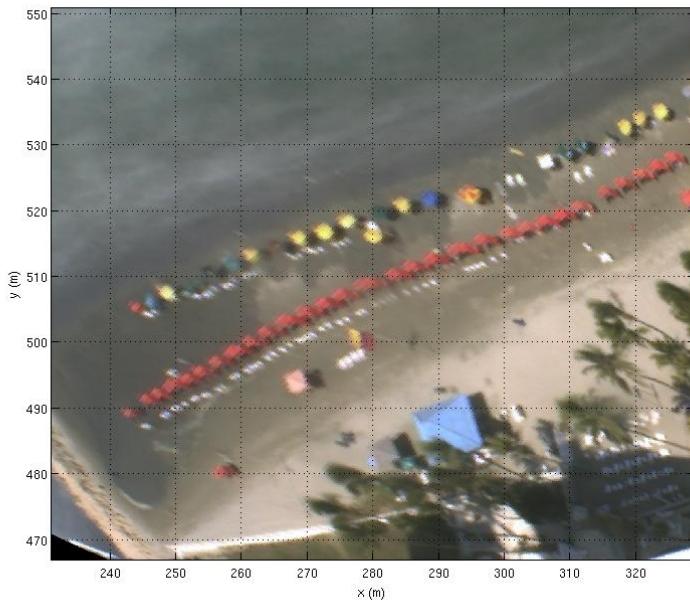


Figura 1.4: Ejemplo de imagen rectificada

de tal manera que es posible medir distancias y tamaños de objetos dentro de la imagen de forma precisa. En la figura 1.4 se muestra un ejemplo de una imagen rectificada.

**Imagen Fusionada** Es la que resulta de unir dos o más imágenes (sean oblicuas o rectificadas) por los puntos en común entre cada par de imágenes consecutivas. Esta fusión se realiza rotando y distorsionando las imágenes de tal manera que queden acopladas en sus puntos comunes, como se muestra en la figura 1.5.



Figura 1.5: Ejemplo de imagen fusionada

**Imagen Snap o Snapshot** Una imagen *snapshot* es aquella que es capturada por una cámara de manera instantánea. Estas imágenes son útiles para obtener una vista estática de un

evento dado, y mediante una secuencia de estas imágenes en el tiempo se puede estudiar la evolución de un fenómeno. En la figura 1.6 se muestra un ejemplo.



Figura 1.6: Ejemplo de imagen *snap*

**Imagen *Timex*** Es el resultado de promediar una secuencia de imágenes capturadas a una frecuencia determinada de antemano. Estas imágenes son útiles cuando se quiere generalizar el movimiento, es decir, los objetos que tienen mayor variabilidad dentro de la imagen se muestran suavizados. Por ejemplo, en la figura 1.7 se muestra una imagen resultado de promediar 120 imágenes a una frecuencia de captura de 2 imágenes por segundo, donde el movimiento de las olas, que es el que tiene mayor variabilidad es promediado.



Figura 1.7: Ejemplo de imagen *timex*

**Imagen *Variance*** Es el resultado de aplicar la desviación estándar de una secuencia de imágenes capturadas a una frecuencia determinada de antemano. La principal característica de estas imágenes, es que lo que está en movimiento se muestra en la imagen de color claro y lo que está estático se muestra en color oscuro. Esto permite resaltar lo que tiene mayor variabilidad en la imagen. En la figura 1.8 se muestra un ejemplo de imagen *variance*.



Figura 1.8: Ejemplo de imagen *variance*

**Resolución** Para las imágenes rectificadas, es la resolución teórica deseada de modo que el tamaño de la imagen estará definido por ella y el tamaño del área a rectificar en el espacio. Por ejemplo, una región de  $10m \times 10m$  corresponderá a una imagen de  $50\text{pixels} \times 50\text{pixels}$  si la resolución elegida es  $1/5(m/pix)$ . La resolución de las imágenes oblicuas no es constante debido a que el plano de la imagen no es paralelo a la región de interés o, en otras palabras, la cámara se encuentra inclinada con respecto a dicha región. Como se puede observar en la figura 1.9, la cual es un mapa de resolución obtenido de una imagen oblicua, es posible ver que la resolución de la imagen disminuye a medida que el área de interés está más alejada de la cámara puesto que cada pixel contiene información de una región más grande el espacio.

Es importante notar que no es posible obtener una resolución infinitamente alta en la imagen rectificada, porque cuando la resolución de la imagen rectificada es menor que la resolución de la imagen oblicua en una región determinada el resultado es la pérdida de una porción de la información contenida en la imagen, pero cuando la resolución de la imagen rectificada es mayor que la resolución de la imagen oblicua, lo que se está haciendo es forzar que los pixeles se estiren, lo que simplemente significa que se usa información redundante para crear la imagen. Por este motivo, la resolución usada en la rectificación no debería ser mayor que la resolución media de la imagen, así no se pierde demasiada información pero la imagen rectificada tampoco tendrá un gran tamaño, lo cual no es práctico.

**Rectificación** Es el proceso mediante el cual se transforma una imagen oblicua como la mostrada en la figura 1.3 en una imagen como la mostrada en la figura 1.4. En las imágenes oblicuas los objetos en el espacio tienen distinta resolución a medida que se alejan o acercan al lente de la cámara. Debido a esto, es imposible hacer mediciones en estas imágenes, tales como hallar distancias, o determinar el tamaño de un objeto.

En una estación de monitorización se tienen varias cámaras que le toman fotos a determinadas regiones de la zona de estudio en intervalos regulares de tiempo definidos de antemano. En general, es necesario convertir las imágenes oblicuas en imágenes rectificadas para hacer mediciones de algún tipo y extraer información útil de ellas, aplicando métodos de rectificación basados en un conjunto de GCPs. Los métodos con los que se cuenta son *DLT* [1, 2, 3, 4] (Transformada Lineal Directa), *RANSAC-DLT* [7] y *Pinhole*

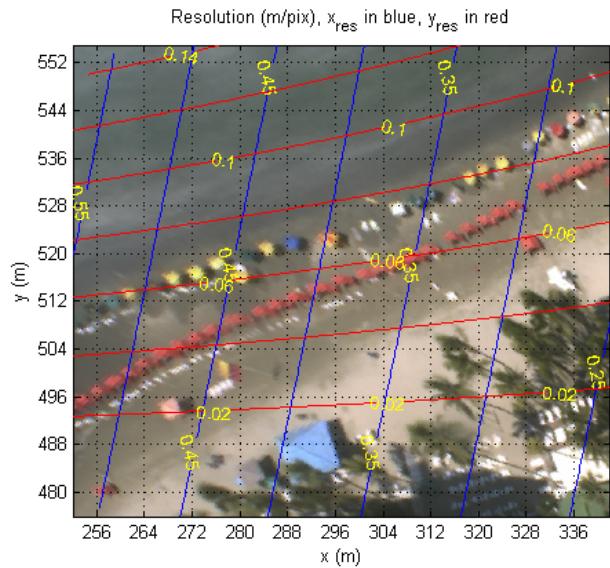


Figura 1.9: Mapa de resolución de una imagen oblicua

[2, 5, 6, 3].

El proceso de rectificación de imágenes se esquematiza en la figura 1.10.

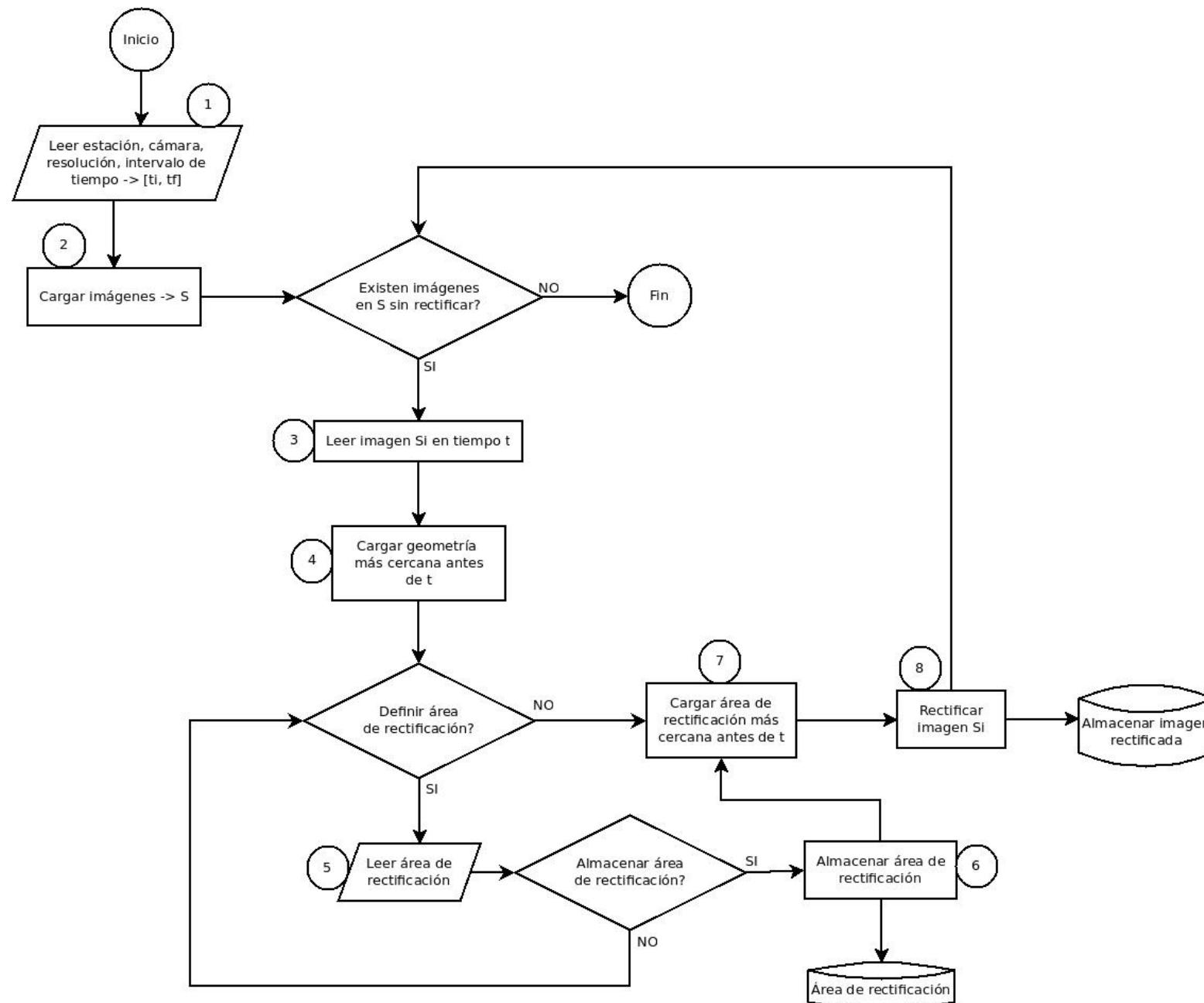


Figura 1.10: Diagrama de flujo de la rectificación en HORUS

Los pasos en el proceso de rectificación son los siguientes:

1. El usuario ingresa la estación, cámara, resolución e intervalo de tiempo en el cual se van a rectificar imágenes.
2. Se cargan todas las imágenes que correspondan a este intervalo de tiempo en el conjunto S y que correspondan a la estación y cámara que se ingresaron.
3. Mientras hayan imágenes sin rectificar en el conjunto de imágenes S, se lee cada imagen.
4. Se carga la calibración de rectificación más cercana antes del tiempo de la imagen actual.
5. Si el usuario decide definir el ROI de rectificación en la imagen, se lee el ROI de rectificación manualmente. Este proceso se repite tantas veces como se requiera.
6. Cuando el usuario está conforme con el ROI de rectificación, se almacena.
7. Se carga el ROI de rectificación más cercano antes del tiempo de la imagen.
8. Se rectifica la imagen teniendo en cuenta los parámetros de rectificación y el ROI de rectificación y se almacena la imagen rectificada.

**Fusión** En el proceso de fusión se generan los parámetros que sirven para unir dos imágenes a través de unos puntos conocidos en común. En la figura 1.11 se esquematiza el proceso para generar una fusión entre dos imágenes. Cuando se calibra la fusión se llevan a cabo los siguientes pasos: Primero, se elige un conjunto de puntos comunes entre las dos imágenes (en la figura 1.11 estos puntos corresponden al triángulo tanto dentro de la imagen de la izquierda como en la imagen de la derecha); se genera una matriz de transformación que es la que permite unir estos puntos en común rotando y/o distorsionando una de las imágenes para que coincida con la otra imagen en sus puntos comunes. Finalmente, las dos imágenes se unen en una sola.

En HORUS se cuenta con dos métodos para estimar las matrices de transformación, cuando la fusión es para imágenes oblicuas: transformación afín y transformación proyectiva. La primera se emplea cuando la diferencia entre las imágenes es un pequeño movimiento de la cámara, mientras que la transformación proyectiva es buena tanto para estos pequeños movimientos como para diferencias más grandes en la posición y la orientación de las cámaras. En ocasiones, la transformación afín puede dar como resultado mejores imágenes fusionadas que la transformación proyectiva.

Para la transformación afín se requieren mínimo tres puntos comunes, mientras que para la transformación proyectiva se requieren mínimo cuatro puntos en común. Cuando el número de puntos en común es alto, se puede optimizar el cálculo de la matriz de transformación mediante el método de optimización *Levenberg–Marquardt*.

En el sistema HORUS se capturan imágenes de la zona de estudio que además de ser rectificadas también pueden ser fusionadas, dado que cada par de imágenes de cámaras consecutivas tengan puntos en común. El usuario puede marcar puntos en un par de imágenes que corresponden a las mismas coordenadas geográficas. Estas marcaciones sirven como insumo para generar los parámetros del modelo que fusiona las imágenes.

El proceso de fusionar imágenes en HORUS se esquematiza en la figura 1.12.

Los pasos que se llevan a cabo en el proceso de fusión son los siguientes:

1. El primer paso es cargar el orden de las cámaras para la fusión. Cuando se definen los parámetros de una fusión, se debe definir cuáles son las cámaras que participan

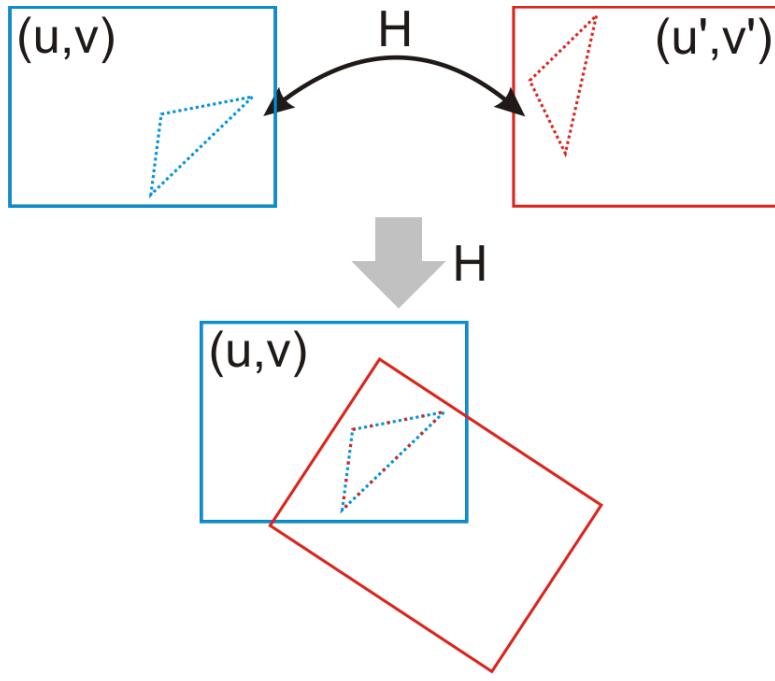


Figura 1.11: Esquema de la fusión entre dos imágenes

en la fusión y en qué orden, esto es porque el método utiliza este orden para fusionar las imágenes de cada cámara.

2. La fusión se realiza en un intervalo de tiempo especificado por el usuario. Se buscan todas las imágenes en este intervalo. El error corresponde al margen para buscar imágenes coincidentes para todas las cámaras en un tiempo específico, ya que es posible que las imágenes difieran por unos pocos segundos.
3. Para todas las imágenes que estén en el intervalo de tiempo, se lee cada una para ser procesada secuencialmente.
4. Se verifica que para el instante de tiempo (+/- error) de cada imagen existan imágenes para todas las cámaras presentes en el orden. Si hay imágenes para todas las cámaras, se agregan a la lista.
5. Si se especificó que las imágenes se rectifican antes de fusionar, para cada imagen en la lista de conjuntos de imágenes se carga la calibración más cercana antes del tiempo de las imágenes, para rectificarlas.
6. Se rectifican las imágenes con los parámetros cargados, y se reemplazan las imágenes originales con la rectificadas.
7. Después de que se rectifican las imágenes para el tiempo actual, si se especificó que se rectificaran, o no, se procede a fusionar las imágenes. Se cargan los parámetros de la fusión más cercanos antes del tiempo de las imágenes.
8. Se fusionan todas las imágenes en el conjunto de imágenes y se almacenan. Mientras hayan más imágenes por procesar, se repite el proceso.

**Timestamp** Representa el tiempo de un objeto o evento que contiene la fecha y hora en un formato específico. Para el resto de este manual, este tiempo es representado por un valor numérico, el número de días desde enero 1 del año 0 más 1. Éste es el mismo formato que utiliza la función `datenum` de MATLAB.

**Hosting o almacenamiento web** Es un servicio que ofrecen algunas compañías en Internet

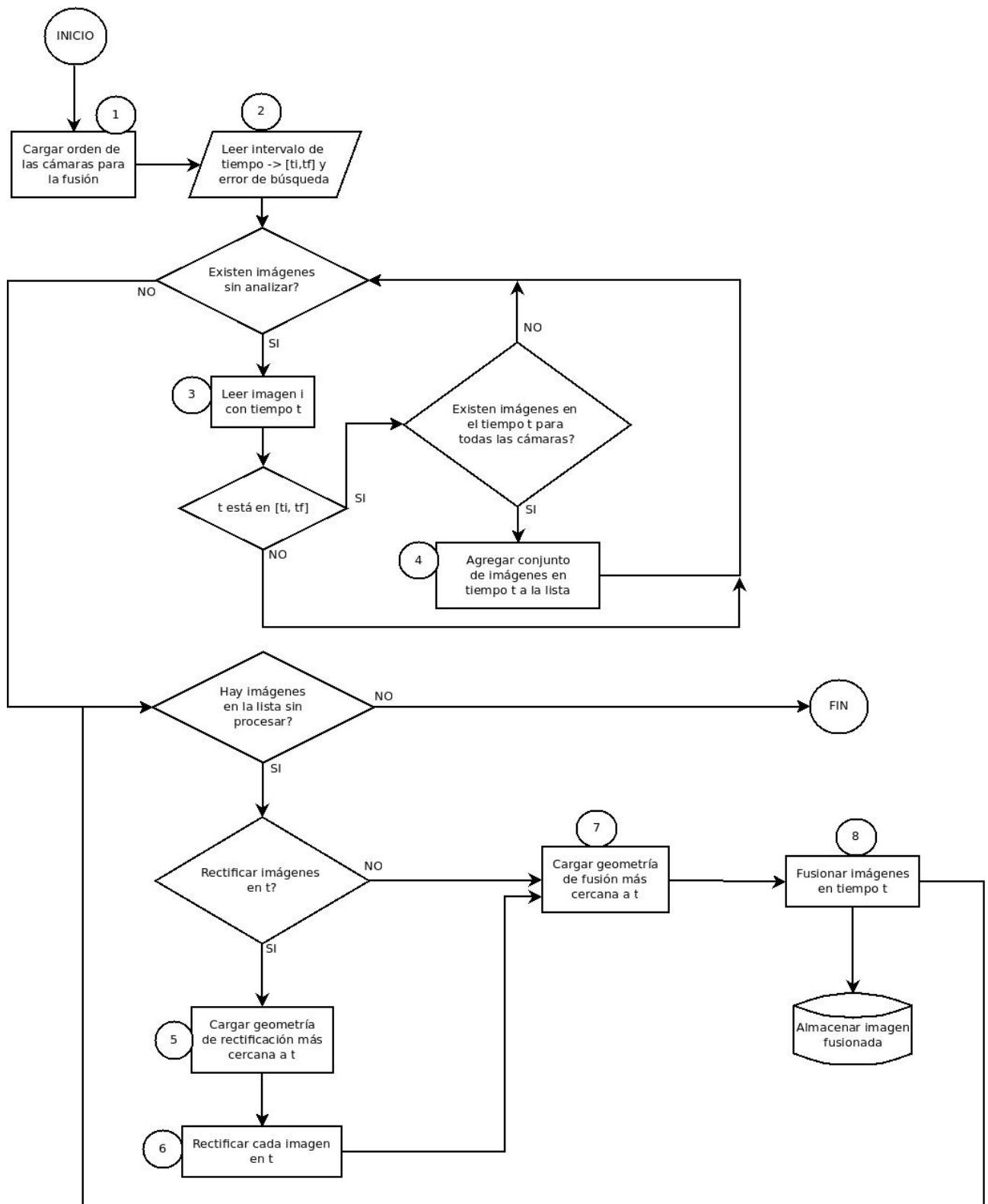


Figura 1.12: Diagrama de flujo de la fusión en HORUS

donde ceden un espacio en sus servidores para subir (alojar) un sitio web para que pueda ser accedido de forma online.

**Servidor de archivos** Es un computador que almacena varios tipos de archivos, en este caso imágenes, y puede ser accedido desde otros computadores de forma online que reciben el nombre de clientes.

**Base de datos satélite** Una base de datos es una colección de información organizada de forma que un software pueda seleccionar rápidamente los fragmentos de datos que necesite. Se le asigna el nombre de base de datos satélite a aquella que sólo posee información de una o más estaciones pero no de todas.

**Base de datos central** Se le asigna el nombre de base de datos central a aquella que posee la información de todas las bases de datos satélites.

**Base de datos hosting** Se le asigna el nombre de base de datos *hosting* a aquella que se encuentra en el *hosting* y posee la información de la base de datos central, y hace las veces de respaldo en un servidor externo.

**Timestack** Es una matriz de series temporales de intensidad de pixel de una región de interés, en palabras coloquiales se puede definir como una secuencia de imágenes de una región de interés en el tiempo a cierta frecuencia de muestreo. En la figura 1.13 se ve como se selecciona una región de interés de un timestack en *cross-shore* o *alongshore* (línea roja y línea amarilla respectivamente). En HORUS los *timestacks* se capturan y almacenan como videos que contienen la secuencia de imágenes.

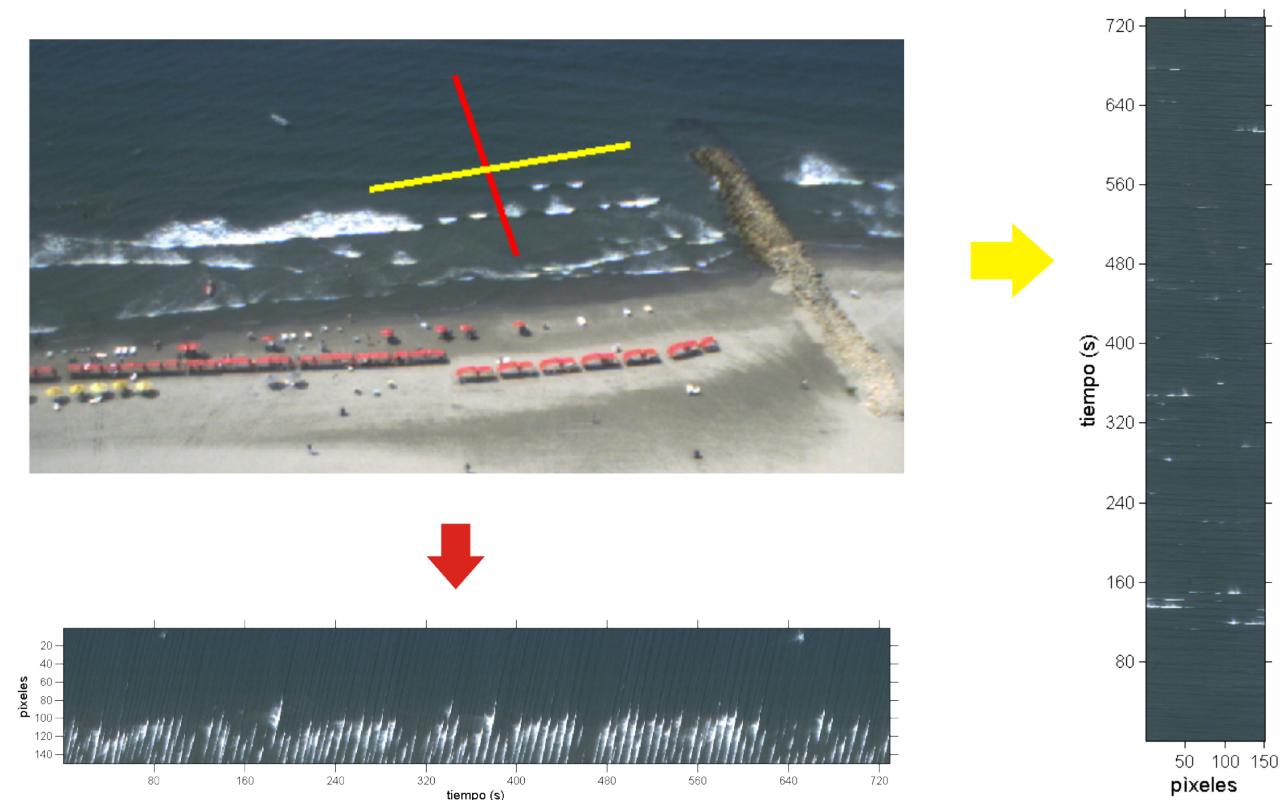


Figura 1.13: Selección de una región para un timestack

**Frame o fotograma** Es una imagen particular dentro de una sucesión de imágenes que conforman un video, al hablar de número de frames se hace referencia a cuantas imágenes va a tener el video.

# Capítulo 2

## Introducción

HORUS es un sistema de monitorización ambiental que consiste en una o varias estaciones ubicadas en lugares específicos, donde hay cámaras que toman fotos periódicamente desde distintos ángulos a una región de la zona de estudio. Las imágenes tomadas por las cámaras son procesadas mediante algoritmos avanzados con el fin de extraer información útil de ellas.

El sistema HORUS está compuesto por los módulos de captura, procesamiento y visualización en la web, como se muestra en la figura 2.1. Cada una de estas componentes tiene asociado un conjunto de funciones escritas en lenguaje MATLAB para las tareas de procesamiento, almacenamiento y extracción de información. Estas funciones están ligadas a una base de datos satélite que almacena toda la información necesaria para procesar las imágenes y proveer información útil tanto para gestores, y científicos como para investigadores. En la figura 2.2 se explica el funcionamiento de HORUS con las bases de datos, en esta figura cada servidor de captura será configurado con un  $\Delta t$  que indica cada cuánto tiempo se hará la captura y la transferencia de la información, éste será escogido por los administradores de la estación dependiendo de sus necesidades. El  $\Delta t$  puede ser diferente o no para todos los procesos de captura, transferencia de información y procesamiento de las imágenes, estos procesos son independientes entre sí.

En el proceso de transferencia de información se envían las imágenes a un servidor de archivos, y se envía la información de la imagen tal como su nombre, la ruta relativa donde será guardada, la fecha de captura, entre otros a una base de datos satélite. El servidor de archivos y la base de datos satélite pueden o no estar en el mismo computador. Para calcular los KB (Kilobytes) que se envían al servidor de archivos se debe sumar el peso de todas las imágenes tomadas en una captura, y para calcular los KB enviados a la base de datos se debe multiplicar la cantidad de imágenes tomadas en una captura por 0.5 KB (de acuerdo a los tamaños de los campos definidos en la base de datos). En la etapa de procesamiento lo que se haga va a depender de las opciones que haya seleccionado el administrador de la estación, pero en forma general, se fusionan las imágenes oblicuas y se fusionan las imágenes rectificadas de tipo *snap* y *timex* y se generan miniaturas de estas imágenes. A medida que se van generando las nuevas imágenes, éstas son insertadas en la base de datos satélite, para luego ser enviadas al *hosting*.

Las bases de datos satélites envían información cada cierto tiempo  $t$  (puede ser una vez al día en horas de la noche) a la base de datos central en donde se concentrará toda la información de las bases de datos satélites. La base de datos central enviará su información también cada tiempo  $t$  a la base de datos del *hosting* que hará las veces de respaldo de los datos. Entre el *hosting* y las bases de datos satélites habrá una comunicación permanente ya que el *hosting* consulta estas bases de datos para poder mostrar las imágenes en la página web. En caso de

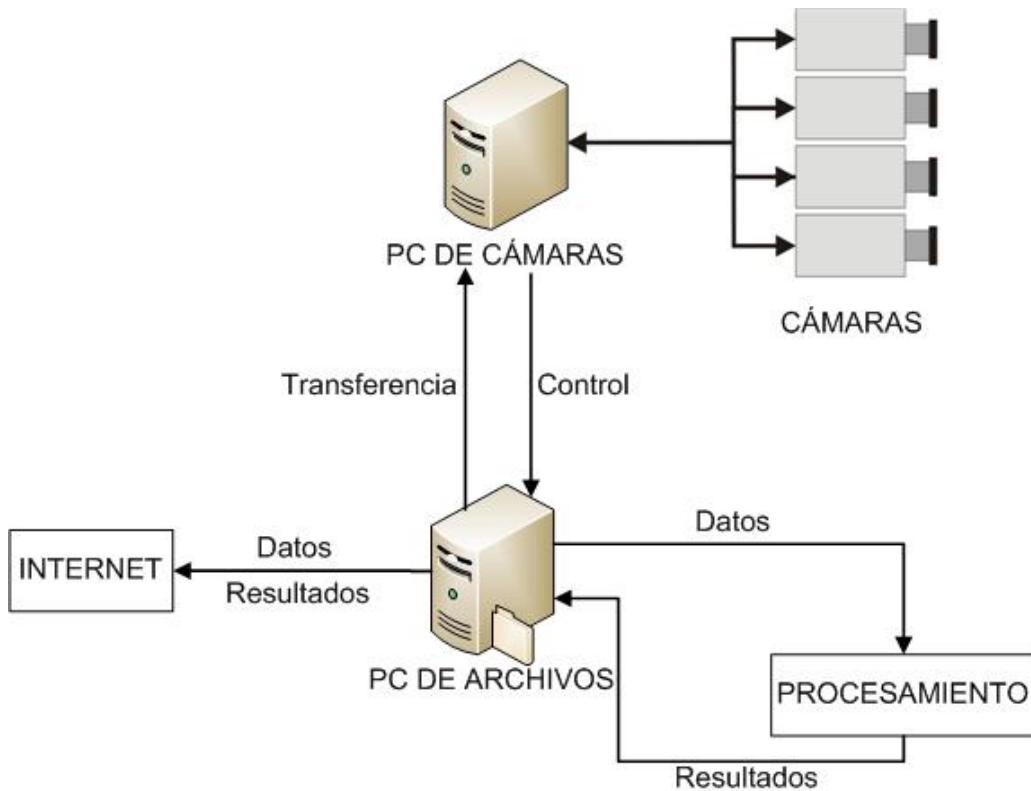


Figura 2.1: Esquema general de HORUS

que haya problemas de conexión entre estos, el *hosting* utilizaría la base de datos que está almacenando para mostrar la información aunque no sea la última generada, por esta razón en la figura la conexión entre estos es punteada.

A continuación se plantea un ejemplo de un caso hipotético para que se entienda mejor el funcionamiento del sistema HORUS. Para empezar se supone que se tiene un sistema de captura que llamaremos CARTAGENA que cuenta con tres cámaras. Éste se comunica con el sistema de archivos A1 y con la base de datos satélite S1. Inicialmente, se define la hora de inicio y fin de la captura de imágenes. Para este ejemplo, serán 6:30 y 18:30, respectivamente. Se define un tiempo  $\Delta t$  para saber cada cuánto se ejecuta la captura, en este caso, 30 minutos y los tipos de imágenes a capturar, en este caso serán *snap*, *timex* y *var*. Para las imágenes *timex* y *var* se debe definir el tiempo de captura, en este ejemplo, 120 segundos.

Se configura la captura de *timestacks*, escogiendo la hora de inicio y fin que será 10:10 y 16:10 respectivamente, el  $\Delta t$  es de 120 minutos y el número de frames es de 1200.

Después de esto, se configura la transferencia de las imágenes capturadas, se comienza escogiendo la hora de inicio y fin que será 6:40 y 18:40 respectivamente y el  $\Delta t$  que será de 30 minutos. Al ejecutarse enviará las imágenes capturadas a A1 e insertará la información de las imágenes en S1.

Por último, se debe configurar el procesamiento en el sistema de archivos A1, en el cual se pondrá la hora de inicio y fin como 6:50 y 18:50, respectivamente, y el  $\Delta t$  que será de 30 minutos. También se configuran las opciones que se quieren procesar, en este caso: fusionar imágenes oblicuas, fusionar imágenes rectificadas de tipo *snap* y *timex*, generar miniaturas de estas imágenes, y se configuran los datos para transferir las miniaturas al *hosting*. Las imágenes

Abreviaciones:  
 BD: Base de datos.  
 $\Delta tCA_i$ : Tiempo de espera entre las capturas para el módulo  $i$  de captura.  
 $\Delta tTA_i$ : Tiempo de espera entre las transferencias para el módulo  $i$  de captura.  
 $\Delta tPA_i$ : Tiempo de espera entre los procesamientos para el módulo  $i$  de captura.  
 KB: Kilobytes.  
 $A_i$ : Información capturada por el módulo  $i$  de captura.  
 $PA_i$ : Información del módulo de captura  $i$  después de ser procesados.  
 $tSi$ : Tiempo en que se sincroniza la BD satélite  $i$  con la BD central.  
 $S_i$ : Información de la BD satélite  $i$ .  
 $tC$ : Tiempo en que se sincroniza la BD central con la BD hosting.  
 C: Información de la BD central.

Nomenclatura  
 Inserta o envía información  
 Trae información de la BD  
 Conexión alternativa a la BD para traer información

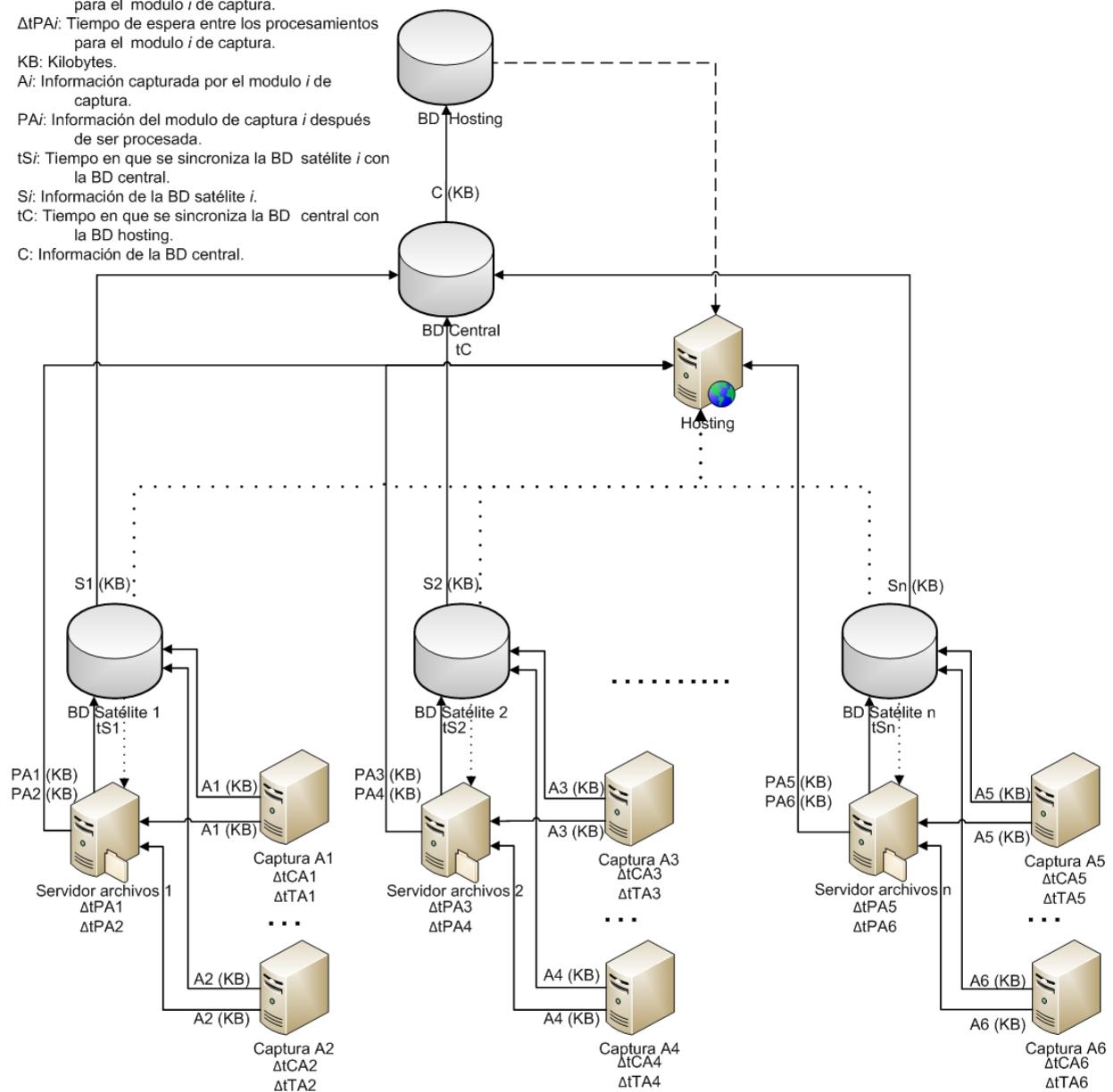


Figura 2.2: Esquema de funcionamiento completo de HORUS

generadas son insertadas en S1 y las miniaturas son enviadas al *hosting*.

Para mayor información de cómo hacer estas configuraciones ver el apartado de automáticos [5.12](#).

En la tabla [2](#) se puede ver el resumen de la información suministrada para configurar la captura.

CARTAGENA				
Información	Captura imágenes	Captura timestamp	Transferencia imágenes	Procesamiento
Hora de inicio	6:30	10:10	6:40	6:50
Hora de fin	18:30	16:10	18:40	18:50
$\Delta t$ (min)	30	120	30	30
Tipos de imágenes	snap, timex y var			snap y timex
Tiempo captura (timex y/o var)	120			
Número de frames		1200		

Cuadro 2.1: Resumen configuración

En la figura [2.3](#) se muestra una línea de tiempo de los procesos antes configurados, con el fin de evaluar si los tiempos son los adecuados y que no haya solapes. Estos tiempos se calculan con las siguientes fórmulas:

- *Captura:*

$$3 * \text{duracion} \text{ [seg]}$$

Para hacer este cálculo se está poniendo una holgura de dos veces la duración de la captura para que el sistema procese las imágenes capturadas y obtener las imágenes de tipo *timex* y *var*.

- *Timestamp:*

$$1.4 * \text{duracion2} \text{ [seg]}$$

Para este cálculo se está poniendo una holgura del 40 % del tiempo de captura de los timestamp para que se procesen las imágenes y se guarde el video.

- *Transferencia:*

$$25 * \text{imagenes} \text{ [seg]}$$

Se supone una velocidad de 20 KB/s y un tamaño por imagen de 500 KB ya que la velocidad se asume baja, en ésta se tiene en cuenta la holgura.

- *Procesamiento:*

$$1.4 * ((29 * \text{imagenes}) + 210) \text{ [seg]}$$

Para el cálculo de este tiempo se hicieron pruebas y se llegó a que el procesamiento dura aproximadamente 29 segundos por cada imagen. El tiempo entre los diferentes procesos es aproximadamente 210 segundos. Se supone una holgura del 40 % para hacer el procesamiento de todas las imágenes.

Donde:

duracion: Tiempo de captura para imágenes tipo timex o var.

duracion2: (numero de frames)/(frame rate), es decir el tiempo de duración del timestamp.

imagenes: Cantidad de imágenes capturadas.

En este manual se describen las interfaces gráficas que acompañan a las funciones que componen el Toolbox, las cuales son una ayuda para el usuario que desea realizar tareas de

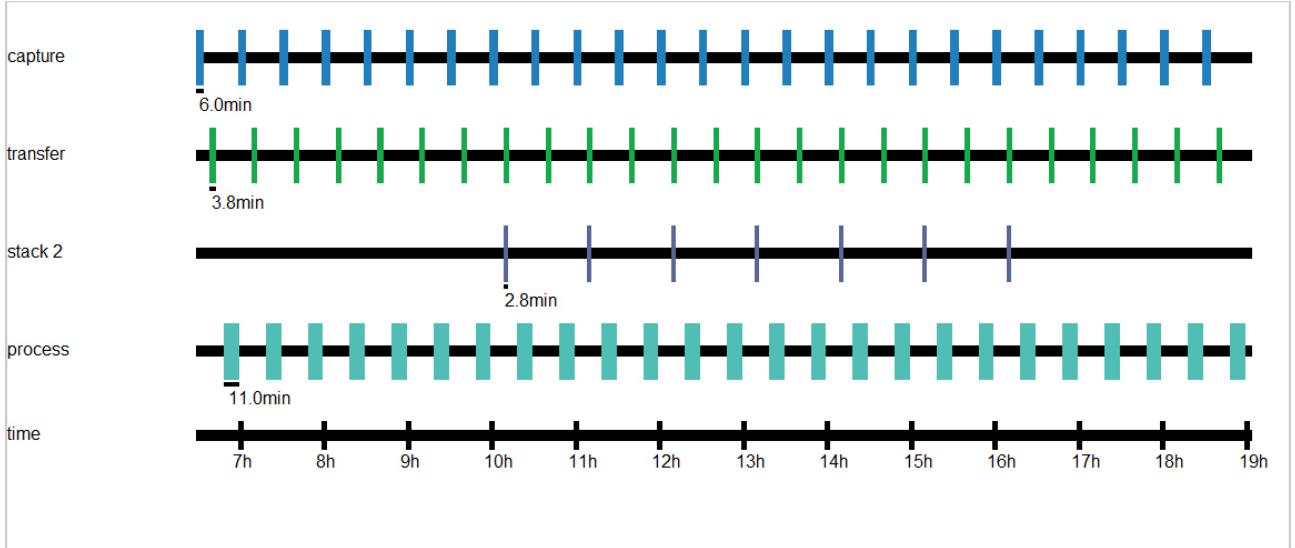


Figura 2.3: Línea de tiempo de los procesos

administración, manipulación y procesamiento de la información de manera asistida, siendo también posible utilizar las rutinas independientemente.

# Capítulo 3

## Diseño de HORUS

### 3.1. Estructura del Toolbox

El sistema HORUS es un paquete con la estructura que se muestra en la figura 3.1.

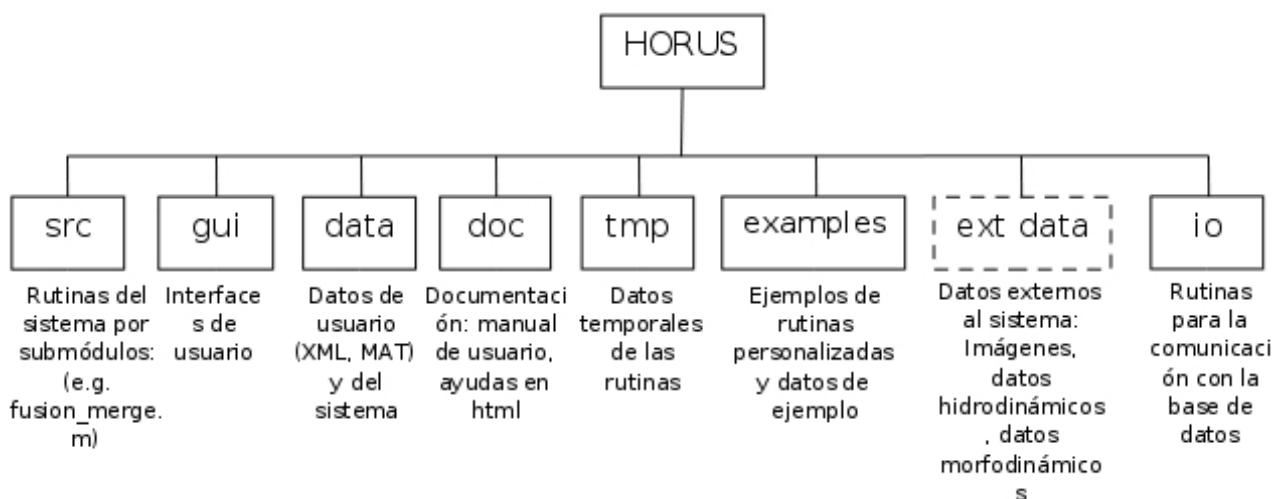


Figura 3.1: Esquema general del sistema de archivos de HORUS

En el esquema se presentan todos los módulos del paquete que contiene el sistema HORUS, los cuales se detallan a continuación:

- Módulo *src*: Este módulo contiene las rutinas en MATLAB que llevan a cabo los procesos de HORUS, por ejemplo, rectificación, fusión, entre otros.
- Módulo *gui*: En este módulo se encuentran las interfaces de usuario, y se hace uso de las rutinas ubicadas en *src* para la funcionalidad y de *io* para la comunicación con las bases de datos.
- Módulo *data*: Contiene los datos de configuración del sistema y del usuario, tales como las rutas de las imágenes, configuraciones, entre otros. Los datos contenidos en este módulo sirven como insumo para la funcionalidad de algunas rutinas y deben estar muy bien estructurados para facilidad de lectura y manipulación.
- Módulo *doc*: Contiene los manuales técnicos y de usuario, así como archivos en diferentes formatos (html, pdf, entre otros) que explican la funcionalidad de submódulos o rutinas.

- Módulo *tmp*: Algunas rutinas requieren generar variables o archivos temporales, en este módulo se almacenan datos que pueden ser desechados posteriormente o no son vitales para el funcionamiento del sistema.
- Módulo *examples*: Este módulo es una extensión de la documentación donde se hacen demostraciones o ejemplos de alguna rutina, y datos de prueba.
- Módulo *ext data*: Este módulo representa la base de datos de imágenes *snap*, *timex*, *var* o *timestacks*, de las zonas de estudio, así como las series de tiempo, e información sobre la estación. La línea punteada del recuadro indica que no se encuentra dentro del paquete de HORUS, pero hay comunicación con este módulo.
- Módulo *io*: Este módulo contiene rutinas específicas para la comunicación con una base de datos. Es el puente entre el módulo *src* y *gui* con la base de datos.

En la figura 3.2 se muestra un diagrama de comunicación entre los distintos módulos según lo explicado anteriormente. El diseño del sistema está constituido por tres capas. La capa de más bajo nivel es la información que se almacena en una base de datos relacional, la siguiente capa es un intermedio entre la aplicación y la base de datos, y está constituida por las funciones que consultan e insertan información en la base de datos desde las funciones de usuario. La capa de más alto nivel, es la que contiene las funciones que el usuario utiliza y las interfaces gráficas.

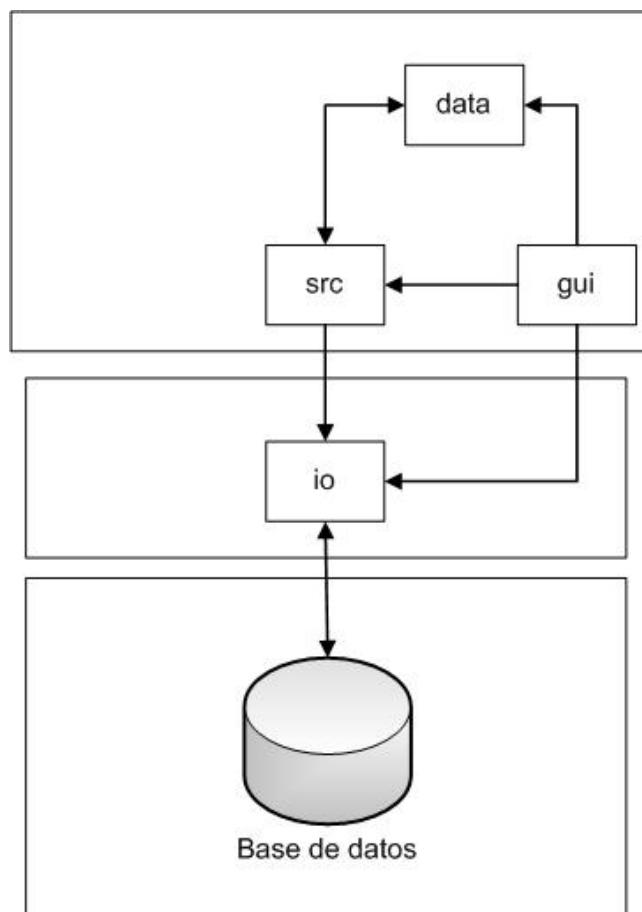


Figura 3.2: Diagrama de comunicación entre los módulos

El sistema HORUS consta de tres componentes básicamente, la captura, que es donde se toman las imágenes y son enviadas a un servidor central, el procesamiento, en el cual se manipulan las imágenes y se extrae información de ellas, y la visualización en la web.

Este manual se enfoca en cada una de las componentes de HORUS, principalmente en la etapa de procesamiento donde los procesos más importantes son la rectificación y la fusión.

## 3.2. Base de datos

### 3.2.1. Modelo Entidad–Relación

La base de datos de HORUS está implementada en MySQL. En la figura 3.3 se muestra el modelo Entidad–Relación (ER) donde se detallan todas las entidades que componen la base de datos, los atributos o características de cada una de estas entidades, y las relaciones entre ellas. En este modelo se ven claramente todas las componentes de la base de datos, las entidades son los recuadros, los atributos son los ítems dentro de los recuadros, y las relaciones se representan con las líneas que unen cada par de entidades.

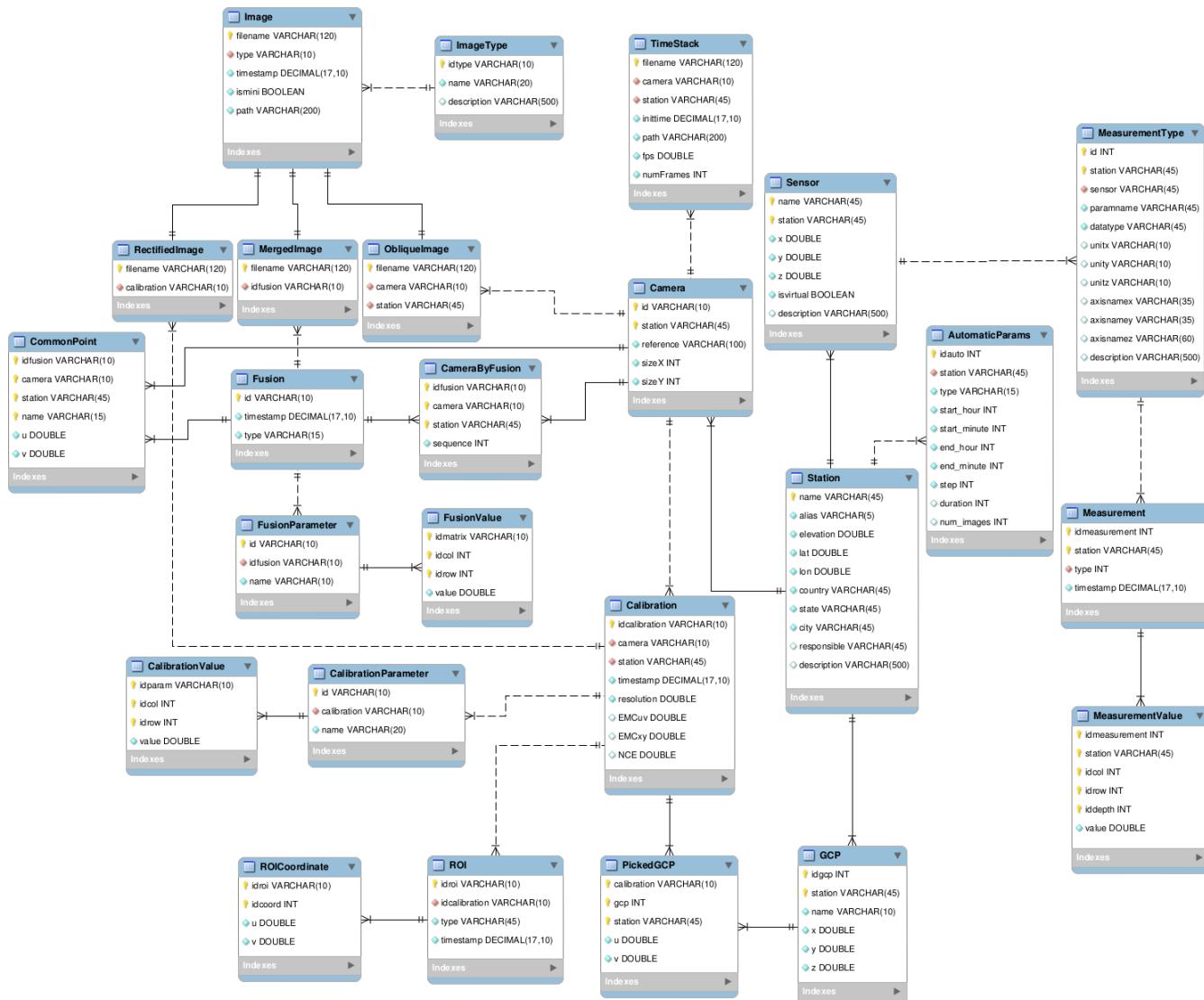


Figura 3.3: Modelo ER de HORUS

### 3.2.2. Entidades

En esta sección se hará una descripción detallada de cada entidad que aparece en el modelo ER de la figura 3.3. *PK* significa clave primaria, el cual es el atributo o conjunto de atributos que identifican de manera única a una instancia de una entidad con respecto a las otras instancias; *FK* significa clave foránea, que es el atributo que referencia a una instancia de otra entidad.

#### Station

Representa a la zona de estudio donde están instaladas las cámaras y se hace la monitorización. Los atributos que contiene son:

- *name (PK)*: Nombre de la estación (e.g. Cartagena, Magdalena). Identifica de manera única al sitio.
- *alias*: Abreviatura del nombre de la estación de máximo 5 caracteres (e.g. CRTG).
- *elevation*: Elevación sobre el nivel del mar en metros (e.g. 15).
- *lat*: Latitud del sitio en grados (e.g. 43,46).
- *lon*: Longitud del sitio en grados (e.g. -3,77).
- *country*: País donde se encuentra ubicada la estación (e.g. Colombia).
- *state*: Estado o departamento donde se encuentra ubicada la estación (e.g. Bolívar).
- *city*: Ciudad donde se encuentra ubicada la estación (e.g. Cartagena).
- *responsible*: Nombre de la entidad o persona responsable de la estación (e.g. Universidad de Cantabria).
- *description*: Descripción en palabras del sitio donde está ubicada la estación.

#### Camera

Esta entidad contiene información sobre las cámaras instaladas en la estación. Los atributos son:

- *id (PK)*: Identificador de la cámara en la estación (e.g. C1, C2).
- *station (PK, FK)*: Nombre de la estación, la cual junto con el identificador de la cámara identifican de manera única a una cámara en la base de datos. Es posible que en diferentes sitios haya cámaras con el mismo *id*.
- *reference*: Marca o modelo de la cámara (e.g. Marlin, Stingray, Sony webcam).
- *sizeX*: Número de píxeles que el sensor puede capturar horizontalmente (e.g. 1024).
- *sizeY*: Número de píxeles que el sensor puede capturar verticalmente (e.g. 768).

## **ImageType**

Esta entidad representa los tipos de imagen que se generan en HORUS: instantáneas (*snapshot*), *time exposure (timex)* y *variance*, y sus características. Los atributos son:

- *idtype (PK)*: Clave que identifica de manera única a cada tipo en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *name*: Nombre del tipo. Sólo se permiten los valores: *snap*, *timex*, *var*.
- *description*: Descripción en palabras del tipo de imagen.

## **Image**

Esta entidad es un supertipo que representa a todas las imágenes que se guardan en la base de datos (oblicuas, rectificadas, fusionadas). No se almacenan las imágenes sino referencias a la localización en un disco duro, o una dirección de red. Los atributos de esta entidad son comunes a todos los tipos de imágenes y son los siguientes:

- *filename (PK)*: Nombre del archivo de la imagen (e.g. 09.12.04.19.00.00.GMT.Cartagena.C1.Snap.1024X768.HORUS.jpg). Se supone que este nombre es único entre todas las imágenes presentes en la base de datos.
- *type (FK)*: Es una clave foránea a la entidad *ImageType*. Representa al tipo de la imagen (e.g. *snap*, *timex*, *var*).
- *timestamp*: El tiempo de la imagen (e.g. 734688,549).
- *ismini*: Atributo booleano que indica si la imagen es o no una miniatura.
- *path*: Localización del archivo de la imagen (e.g. /home/horus/images/).

## **RectifiedImage**

Es un subtipo de *Image*. Representa a las imágenes rectificadas. Sus atributos son:

- *filename (PK, FK)*: Clave foránea a la entidad *Image*.
- *calibration (FK)*: Clave foránea a la entidad *Calibration* y representa la calibración utilizada para rectificar la imagen.

## **MergedImage**

Es un subtipo de *Image*. Representa a las imágenes fusionadas, ya sean rectificadas u oblicuas (este tipo se encuentra en la entidad *Fusion*). Sus atributos son:

- *filename (PK, FK)*: Clave foránea a la entidad *Image*.
- *idfusion (FK)*: Clave foránea a la entidad *Fusion*. Representa la geometría utilizada para fusionar varias imágenes.

## ObliqueImage

Es un subtipo de Image. Representa a las imágenes oblicuas a las que no se les ha aplicado ninguna transformación. Sus atributos son:

- *filename* (*PK, FK*): Clave foránea a la entidad *Image*.
- *camera* (*FK*): Cámara que generó la imagen oblicua. Es clave foránea a la entidad *Camera* (e.g. C2).
- *station* (*FK*): Estación a la que corresponde la imagen. Es clave foránea a la entidad *Camera* (e.g. Cartagena).

## TimeStack

Esta entidad representa a otro tipo de captura que se puede realizar con las cámaras. Un *timestack* se almacena como un vídeo que contiene una secuencia de imágenes capturadas a una determinada frecuencia, durante un tiempo determinado por el número de imágenes que se capturan. Al igual que las imágenes, no se almacena en la base de datos el vídeo como tal sino una referencia a la localización donde se encuentra almacenado, bien sea en un disco duro o en una dirección de red. Sus atributos son:

- *filename* (*PK*): Nombre del archivo del *timestack* (e.g. 11.03.24.12.00.00.GMT.CARTAGENA.C2.STACK.615.0.20X760.HORUS.jpg). Se supone que es único entre todos los *timestacks* presentes en la base de datos.
- *camera* (*FK*): Cámara que generó el *timestack*. Es clave foránea a la entidad *Camera*.
- *station* (*FK*): Estación donde se capturó el *timestack*. Es clave foránea a la entidad *Camera*.
- *inittime*: Tiempo de inicio de la captura.
- *path*: Localización del archivo del *timestack* (e.g. /home/horus/stacks/).
- *fps*: Frecuencia o *framerate* en Hertz. Es el número de imágenes que se capturan por segundo (e.g. 2).
- *numFrames*: Número de frames que se capturan a la frecuencia dada (e.g. 1200).

## GCP

Esta entidad representa a los puntos de control georeferenciados para un sitio que sirven como punto de partida para los modelos de optimización utilizados para rectificar y fusionar imágenes. Sus atributos son:

- *idgcp* (*PK*): Es un identificador de cada punto en un sitio. Es posible que haya dos GCPs en estaciones distintas con el mismo identificador.
- *station* (*PK*): Nombre de la estación a la que corresponde un GCP. Junto con el *idgcp* identifican de manera única un GCP en la base de datos.
- *name*: Nombre que se le da al GCP para facilidad de representación (e.g. GCP001).
- *x, y, z*: Coordenada georreferenciada del GCP (e.g. 500, 1000, 0).

## Calibration

Esta entidad representa una calibración de una cámara en una fecha en particular mediante un modelo como el *DLT* o *Pinhole*. Pueden haber varias calibraciones en el tiempo debido a cambios leves en la posición de las cámaras. Sus atributos son:

- *idcalibration (PK)*: Clave que identifica de manera única a cada calibración en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *camera (FK)*: Cámara en la que se hace la calibración. Es clave foránea a la entidad *Camera*.
- *station (FK)*: Estación donde se hace la calibración. Es clave foránea a la entidad *Camera*.
- *timestamp*: Fecha en la que se hizo la calibración.
- *resolution*: Resolución de la rectificación (e.g. 0,5).
- *EMCuv*: Error cuadrático medio del modelo en píxeles.
- *EMCxy*: Error cuadrático medio de las proyecciones en metros.
- *NCE*: Error de calibración normalizado.

## PickedGCP

Para cada calibración, hay un subconjunto de los GCP de una estación que son escogidos como insumo para el modelo de optimización. Esta entidad representa cuáles GCP que corresponde a una estación están relacionados con una calibración en particular y la relación entre sus coordenadas ( $x, y, z$ ) y las coordenadas ( $u, v$ ) marcadas por el usuario. Los atributos son:

- *calibration (PK, FK)*: Es la calibración a la que están relacionados los puntos. Es clave foránea a la entidad *Calibration*.
- *gcp (PK, FK)*: Es el número del punto de control que se relaciona. Es clave foránea a la entidad *GCP*.
- *station (PK, FK)*: Es la estación a la que corresponde el GCP. Es clave foránea a la entidad *GCP*.
- *u, v*: Posición en píxeles en la cámara de cada punto de control transformado mediante los parámetros de la calibración a la que está relacionado (e.g. 300, 200).

## CalibrationParameter

Esta entidad representa los parámetros que pueden ser matrices o escalares de una calibración. En realidad no se almacena el valor del parámetro, sino su información. Los atributos son:

- *id (PK)*: Clave que identifica de manera única a cada parámetro en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *calibration (FK)*: Es la clave foránea o la referencia a la calibración que está relacionada con el parámetro.
- *name*: Es el nombre del parámetro, es útil para saber exactamente a qué matriz o valor escalar se refiere el parámetro (e.g. H, P, K).

## CalibrationValue

Esta entidad representa la matriz donde se almacena el valor del parámetro de una calibración. La manera de representar a una matriz, ya que el gestor de base de datos no posee un tipo de datos matriz, es como un conjunto de filas que corresponden a cada elemento de la matriz y cada una se identifica mediante un identificador de columna, de fila, y valor. Los atributos son:

- *idparam* (*PK, FK*): Es la clave foránea o referencia a la entidad *CalibrationParameter*, y representa el parámetro al que está relacionado.
- *idcol* (*PK*): Identificador de la columna de la matriz (los índices comienzan en 1).
- *idrow* (*PK*): Identificador de la fila de la matriz (los índices comienzan en 1).
- *value*: Valor escalar de cada elemento de la matriz.

## Fusion

Esta entidad representa los parámetros que son específicos para fusionar imágenes, bien sea rectificadas u oblicuas. Los atributos son:

- *id* (*PK*): Clave que identifica de manera única a los datos de la fusión en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *timestamp*: Tiempo en el que se crearon los parámetros para la fusión. Sirve para encontrar la fusión más cercana a una fecha especificada.
- *type*: Es el tipo de imágenes que se va a fusionar. Los valores que puede tomar son: *rectified* y *oblique*.

## FusionParameter

Al igual que la entidad *CalibrationParameter*, representa una matriz o escalar de la calibración para una fusión o la transformación afín entre dos imágenes. No se almacena aquí el valor. Los atributos son:

- *id* (*PK*): Clave que identifica de manera única a una matriz de fusión en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *idfusion* (*FK*): Clave foránea o referencia a la entidad *Fusion* e identifica la fusión a la que corresponde la matriz.
- *name*: Es el nombre de la matriz (e.g. H12, H23).

## FusionValue

Esta entidad representa la matriz donde se almacena el valor del parámetro de una fusión. La manera de representar a una matriz, ya que el gestor de base de datos no posee un tipo de datos matriz, es como un conjunto de filas que corresponden a cada elemento de la matriz y cada una se identifica mediante un identificador de columna, de fila, y valor. Los atributos son:

- *idmatrix* (*PK, FK*): Es la clave foránea o referencia a la entidad *FusionParameter*, y representa la matriz al que está relacionado.
- *idcol* (*PK*): Identificador de la columna de la matriz (los índices comienzan en 1).
- *idrow* (*PK*): Identificador de la fila de la matriz (los índices comienzan en 1).
- *value*: Valor escalar de cada elemento de la matriz.

## CameraByFusion

Para realizar una fusión es necesario saber cuáles cámaras están involucradas y en qué orden de la fusión va cada cámara. Para representar esto se utiliza esta entidad que almacena para cada fusión las cámaras relacionadas y el orden de cada una. Los atributos son:

- *idfusion* (*PK, FK*): Es la clave foránea o referencia a la fusión a la que se le quieren relacionar las cámaras.
- *camera* (*PK, FK*): Es el identificador de la cámara que se quiere incluir. Es clave foránea a la entidad *Camera*.
- *station* (*PK, FK*): Es la estación en la que se realiza el proceso. Es clave foránea a la entidad *Camera*.
- *sequence*: El orden en el que la cámara actúa en la fusión (comenzando en 1).

## Common Point

Para cada fusión, hay un conjunto de puntos comunes entre cada par de cámaras de una estación que son escogidos como insumo para el modelo de optimización. Esta entidad representa estos puntos, asociados a una cámara de la estación relacionados con una fusión en particular. Los atributos son:

- *idfusion* (*PK, FK*): Es la fusión a la que están relacionados los puntos. Es clave foránea a la entidad *Fusion*.
- *camera* (*PK, FK*): Es el ID de la cámara a la que está asignado cada punto. Es clave foránea a la entidad *Camera*.
- *station* (*PK, FK*): Es la estación a la que corresponde el punto. Es clave foránea a la entidad *Camera*.
- *name* (*PK*): Nombre asignado al punto en común. Este nombre es compartido con el otro punto correspondiente en la otra cámara.
- *u, v*: Posición en píxeles en la cámara de cada punto común (e.g. 300, 200).

## ROI

En HORUS los ROIs tienen diversas aplicaciones, por ejemplo, para definir el área de la imagen que se captura en un *timestack*, el área que se quiere rectificar o fusionar en una imagen, entre otros. La manera de representar un ROI es mediante un conjunto de coordenadas (*u, v*) en píxeles que representan los vértices del polígono. La entidad ROI representa esto, aunque no almacena las coordenadas como tal. Sus atributos son los siguientes:

- *idroi* (*PK*): Clave que identifica de manera única a un ROI en la base de datos. Se representa como el alias de la estación más un autonumérico (e.g. CRTG00023).
- *idcalibration* (*FK*): Es la clave foránea a la entidad *Calibration* y representa la calibración a la que está relacionado el ROI.
- *type*: Es el tipo de ROI o el tipo de aplicación de un ROI en particular. Los valores que puede tomar son: *fusion, rect, stack, user*.
- *timestamp*: Fecha en la que se creó el ROI.

## ROICoordinate

Esta entidad representa las coordenadas de los vértices del polígono que definen un ROI en particular. Los atributos son:

- *idroi* (*PK, FK*): Es la clave foránea al ROI al que pertenece cada coordenada.
- *idcoord* (*PK*): Es un número que identifica cada coordenada. Pueden haber números iguales en distintos ROIs. El polígono se forma uniendo mediante líneas rectas dos puntos con *idcoord* consecutivos. El último se une con el primero.
- *u, v*: El valor de la coordenada en píxeles.

## Measurement

Esta entidad representa las series de tiempo de medición de diversas variables que es interesante mantener almacenadas, por ejemplo, altura de ola, marea astronómica, densidad de usuarios, posición de la línea de costa, entre otros. En esta entidad no se guarda el valor como tal de la medición ya que puede ser un escalar, una matriz de dos o incluso tres dimensiones. Los atributos son:

- *idmeasurement* (*PK*): Autonumérico que identifica de manera única a una medición en la base de datos.
- *station*: Nombre de la estación a la que está relacionado (e.g. Cartagena, Magdalena).
- *type* (*FK*): Tipo de medición. Es clave foránea a la entidad *MeasurementType*.
- *timestamp*: Tiempo de la medición.

## MeasurementType

Esta entidad representa el tipo de una medición en particular. Los atributos son:

- *id* (*PK*): Autonumérico que identifica a un tipo de medición en la base de datos.
- *station (PK FK)*: Estación a la que pertenece la medición.
- *sensor (FK)*: Es la clave foránea a la entidad *Sensor* y representa el sensor con el que se miden las mediciones de un tipo en particular.
- *paramname*: Es el nombre del parámetro (e.g. Hs, T).
- *datatype*: Es el tipo de dato que se esta guardando. Los valores que puede tomar son: *series, matrix*.
- *unitx*: Unidad de medida del eje x en el tipo de medición (e.g. m, °C, m/s).
- *unity*: Unidad de medida del eje y en el tipo de medición (e.g. m, °C, m/s).
- *unitz*: Unidad de medida del eje z en el tipo de medición (e.g. m, °C, m/s).
- *axisnamex*: Es el mensaje que se colocaría en el eje x de una gráfica.
- *axisnamey*: Es el mensaje que se colocaría en el eje y de una gráfica.
- *axisnamez*: Es el mensaje que se colocaría de titulo de una gráfica.
- *description*: Descripción verbal del tipo de medición.

## MeasurementValue

Esta entidad representa el valor de una medición. Una medición se puede representar como un valor escalar, una matriz de dos o tres dimensiones. Cada valor de la matriz se identifica por la medición a la que corresponde, un identificador de fila, columna y profundidad. Los atributos son:

- *idmeasurement (PK, FK)*: Clave foránea o referencia a la medición a la que corresponde cada elemento de la matriz.
- *station (PK FK)*: Estación a la que pertenece la medición.
- *idcol (PK)*: Identificador de la columna de la matriz (los índices comienzan en 1).
- *idrow (PK)*: Identificador de la fila de la matriz (los índices comienzan en 1).
- *iddepth (PK)*: Identificador de la profundidad (tercera dimensión) de la matriz (los índices comienzan en 1).
- *value*: Valor del elemento de la matriz.

## Sensor

Esta entidad representa la información de un sensor con el que se mide algún tipo de medición (e.g. Boya es un sensor para medir altura de ola, período de la ola, entre otros). Los atributos son los siguientes:

- *name*: Nombre único del sensor (e.g. DCX22).
- *station (PK FK)*: Nombre de la estación a la que está relacionado el sensor. Es clave foránea a la entidad *Station*.
- *x, y, z*: Coordenada georreferenciada del sensor (e.g. 500, 1000, 0).
- *description*: Descripción verbal del sensor.

## AutomaticParams

Esta entidad contiene la configuración de los tiempos de los automáticos para una estación. Con estos tiempos, es posible sincronizar procesos como la rectificación y fusión automáticas. Los atributos que contiene son:

- *idauto (PK)*: Autonumérico que identifica de manera única a un automático en la base de datos.
- *station (FK)*: Estación a la que pertenece la captura.
- *type*: Tipo de la captura, puede ser *image, stack, transfer* o *process*.
- *start\_hour*: Hora de inicio de la captura en el día (e.g. 06).
- *start\_minute*: Minuto de inicio de la captura en una hora (e.g. 30).
- *end\_hour*: Hora de finalización de la captura en el día (e.g. 18).
- *end\_minute*: Minuto de finalización de la captura en una hora (e.g. 30).

- *step*: Tiempo de espera entre captura y captura, en minutos.
- *duration*: Tiempo de duración de la captura.
- *num\_images*: Numero de imágenes capturadas.

# Capítulo 4

## Instalación

### 4.1. Instalación Básica

El sistema HORUS fue diseñado para trabajar con la versión de ®MATLAB R2011b o superior y requiere el Toolbox de Procesamiento de Imágenes y el de Bases de datos. Si su instalación de ®MATLAB cumple los requisitos básicos puede poner los archivos del sistema HORUS en cualquier directorio (e.g. `/home/horus`). Los archivos del sistema HORUS inicialmente están comprimidos en `horus.zip`, el primer paso es descomprimirlos en el directorio de preferencia. El computador donde se ejecute el sistema HORUS debe contar mínimo con 2GB de memoria RAM. Si la captura y el procesamiento de las imágenes se hacen en máquinas diferentes, es necesario tener un servidor SSH en el computador donde funcione la captura para enviar la información al computador de procesamiento. Si se tiene un sitio web para enviar información (e.g. [www.horusvideo.com](http://www.horusvideo.com)), el sitio debe tener servicio de FTP para recibir la información desde el computador de procesamiento.

Para visualizar todas interfaces gráficas apropiadamente, se recomienda una resolución de pantalla de mínimo  $1200 \times 800$ .

### 4.2. Base de Datos

El sistema HORUS fue diseñado para trabajar con una base de datos relacional en MySQL. Antes de trabajar con HORUS es necesario configurar la base de datos ya que en ésta va a estar almacenada toda la información que necesita HORUS para trabajar. Es posible trabajar con el sistema HORUS a partir de una base de datos existente a la que se pueda acceder, bien sea mediante una red local o por Internet. También es posible configurar una nueva base de datos desde cero.

Este paso se puede realizar utilizando la interfaz gráfica *Setup database* de la interfaz principal (ver Capítulo 5), o realizando los pasos manualmente. Se recomienda utilizar la interfaz gráfica mientras sea posible, se deja la configuración manual para usuarios avanzados.

#### 4.2.1. Configuración mediante la interfaz gráfica *Setup database*

La interfaz *Setup database* se puede ejecutar presionando el botón correspondiente en la interfaz principal de HORUS o escribiendo el comando `gui_configure_database` en la línea de comandos de Matlab. En la figura 4.1 se muestra la interfaz.

Figura 4.1: Interfaz para configuración inicial de la base de datos

Las componentes numeradas en la interfaz se explican a continuación (los campos con \* son obligatorios):

1. Database host: Es la dirección IP o nombre del host donde se encuentra el servidor MySQL.
2. Database name: Todas las bases de datos deben tener un nombre. En este campo se especifica este nombre, el cual es diferente para las bases de datos presentes en el servidor.
3. Database port: Es el puerto del servidor MySQL, a menos que el administrador del servidor haya decidido seleccionar uno diferente, el puerto por defecto es 3306.
4. Admin username: Es el nombre de usuario del administrador de la base de datos de HORUS. Este usuario es el encargado de realizar las modificaciones principales en la base de datos y de crear nuevos usuarios con privilegios limitados.
5. Admin password: Es la contraseña del usuario administrador de la base de datos.
6. Database SQL file: Se debe seleccionar el archivo en formato SQL con la especificación del esquema relacional de HORUS. Este archivo se provee junto con el software en `examples/horus.sql`. Sin embargo, es posible definir otro esquema siempre y cuando cumpla con las especificaciones de la base de datos, con datos insertados en las tablas de la base de datos.
7. Apply configuration: Este botón ejecuta la configuración creando el esquema de la base de datos en el servidor MySQL y el usuario administrador. Desde aquí es posible insertar información en la base de datos utilizando el usuario administrador.

#### 4.2.2. Configuración manual

##### Configuración del cliente de MySQL

Para establecer la conexión a una base de datos MySQL existente, es necesario configurar MATLAB para trabajar como cliente. En MATLAB existe una variable de entorno llamada `$matlabroot` que representa el directorio de instalación de MATLAB (e.g. `C:\MATLAB\R2011b`), y la utilizaremos para localizar los archivos propios de MATLAB. Los pasos para lograr esto son los siguientes:

1. Asegurarse que el *Database Toolbox* de MATLAB esté instalado junto con MATLAB.
2. Descargar el *plugin* para conexión con MySQL. En el momento de escribir este manual, este plugin se puede descargar desde <http://dev.mysql.com/downloads/connector/j> seleccionando el sistema operativo correspondiente.
3. Descomprimir el archivo.
4. En el directorio descomprimido hay un archivo llamado `mysql-connector-java-<version>-bin.jar` (donde `<version>` es la versión del plugin). Copiar este archivo al directorio `$matlabroot/java/jarext/`.
5. Agregar al archivo `$matlabroot/toolbox/local/classpath.txt` la línea `$matlabroot/java/jarext/mysql-connector-java-<version>-bin.jar` (donde `<version>` es la versión del plugin). En el sistema operativo Windows 7 este archivo no se puede editar por la ubicación en la que se encuentra por esta razón se debe

copiar este archivo en otra carpeta como por ejemplo en el escritorio, hacer el cambio y reemplazar el archivo anterior. Por último si MATLAB se encontraba ejecutándose para que los cambios funcionen se debe de cerrar y volver a abrir.

Ahora, es posible que las funciones de MATLAB hagan conexión con una base de datos existente. Se debe establecer la dirección de conexión del servidor de la base de datos.

## Configuración de una nueva base de datos

Las funciones de HORUS utilizan una base de datos cuya estructura se detalla en el capítulo 3.2. Para crear una nueva base de datos con esta estructura se deben seguir los siguientes pasos (si ya existe una instalación anterior de MySQL, no es necesario volver a instalarlo ya que puede generar problemas debido a cuentas de usuario anteriores):

1. Instalar el servidor de MySQL, que se puede descargar desde <http://dev.mysql.com/downloads/mysql/>. En los pasos de la instalación se deben tener en cuenta los siguientes aspectos en Windows:

- Seleccionar instalación *Típica*.
- Seleccionar la opción “Instalar como *Servicio del sistema*”.
- Seleccionar la opción “Incluir la ruta de la instalación en el PATH del sistema”.
- Reiniciar PC.
- Por defecto, existe un usuario administrador del servidor. Se debe especificar la contraseña de este usuario (**root**).

Si se quiere instalar el servidor de MySQL en Linux, es suficiente con instalarlo desde el repositorio de paquetes de la distribución específica.

2. Inicialmente, el único usuario en MySQL es el usuario **root**. En la línea de comandos acceder a MySQL mediante el comando:

```
mysql -h <dirección> -u root -p
```

Donde <dirección> es la dirección IP del servidor MySQL (**localhost** si es local). Crear la base de datos mediante el comando:

```
mysql> CREATE DATABASE horus;
```

El nombre de la base de datos en este caso es **horus**, sin embargo, este nombre es arbitrario, puede ser cualquiera.

3. Se deben crear los usuarios administradores de la base de datos, que tendrán acceso a la base de datos y sus permisos. Supongamos que queremos crear el usuario **horususer** que tendrá todos los privilegios para manipular la base de datos. Los comandos para crear el usuario con todos los privilegios, son los siguientes:

```
mysql> GRANT ALL PRIVILEGES ON horus.* TO 'horususer'@'%'  
      IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> FLUSH PRIVILEGES;  
mysql> QUIT;
```

Donde <password> es la contraseña del usuario.

4. Ya hemos creado un usuario y un esquema de la base de datos, ahora debemos crear la estructura de la base de datos. El archivo `horus.sql` del directorio `examples` en el paquete de HORUS contiene el código SQL para crear la estructura de la base de datos. Para ejecutar este código se ingresa el siguiente comando desde la línea de comandos del sistema (suponiendo que está en el directorio `examples`):

```
mysql -h <dirección> -u horususer -p horus < horus.sql
```

Donde <dirección> es la dirección IP del servidor MySQL (`localhost` si es local).

*Precaución:* Al ejecutar este comando, toda la información previa guardada en la base de datos `horus` será eliminada (en caso de una instalación anterior).

Opcionalmente, si se dispone de un archivo de inserciones en SQL de la nueva información, podemos utilizar este archivo para alimentar la base de datos, en caso contrario, es necesario insertar los datos manualmente, mediante las interfaces gráficas descritas en el capítulo 5. Supongamos que tenemos el archivo `insert.sql` que contiene la información nueva de la base de datos. Para insertar toda la información en la base de datos, se ejecuta el siguiente comando desde la línea de comandos del sistema (suponiendo que está en el mismo directorio donde está el archivo `insert.sql`):

```
mysql -h <dirección> -u horususer -p horus < insert.sql
```

Cuando se quiere utilizar manualmente alguna de las funciones que se comunican con la base de datos, del directorio `io`, es necesario crear la conexión con la base de datos de antemano. El siguiente código sirve para crear una nueva conexión:

```
try
    conn = connection_db();
catch e
    disp(e.message)
end
```

Si no se ha iniciado una sesión en HORUS, se solicitan los datos de *login*. Es importante que luego de utilizar alguna función de `io`, se cierre la conexión con la base de datos. Esto se puede hacer así:

```
close(conn)
```

# Capítulo 5

## GUI

Las secciones que siguen contienen una descripción de las interfaces gráficas usadas en HORUS para llevar a cabo los diferentes procesos como configuración de la base de datos, configuración de la captura, procesamiento de las imágenes y almacenamiento en la base de datos. Estas interfaces se encuentran en el directorio `gui`.

Todas las interfaces que serán descritas pueden ser llamadas desde la interfaz principal de HORUS (`gui_main`), como se muestra en la figura 5.1.

Figura 5.1: Interfaz principal de HORUS

Para ejecutar la interfaz principal desde Matlab, ubicarse en el directorio principal de HORUS y ejecutar el siguiente comando:

```
> gui_main
```

Cuando se ejecuta por primera vez alguna de las interfaces que se detallan a continuación (excepto la interfaz *Setup database*), se despliega una interfaz para hacer el *login* en la base de datos, con el usuario creado previamente en la instalación. En la figura 5.2 se muestra la ventana de *login*. Este *login* solamente se hace una vez mientras dura la sesión de usuario. Cuando se cierra alguna interfaz, se pregunta al usuario si desea cerrar la sesión o no.

Las componentes numeradas de la interfaz son:

1. Nombre de usuario de la base de datos.
2. Contraseña (no se muestran los caracteres mientras se digitan).
3. Nombre o dirección del servidor donde se encuentra alojada la base de datos (`localhost` si es local).
4. Nombre del esquema de la base de datos (por defecto es `horus`).
5. Puerto de la base de datos (por defecto es 3306).

*NOTA:* Es posible que cuando se esté utilizando una interfaz gráfica, la conexión con la base de datos caduque, por lo que en ese caso, hay que cerrar y abrir nuevamente la interfaz con el fin de renovar la conexión.

El manejo de sesiones en HORUS no tiene que ver con la conexión a la base de datos y la vigencia de ésta, sino con la información del archivo `userinfo.dat` del directorio `tmp`. Este archivo de texto plano contiene el nombre de usuario de la base de datos (`user`), la contraseña encriptada (`password`), la dirección del servidor de la base de datos (`host`), el puerto (`port`) y

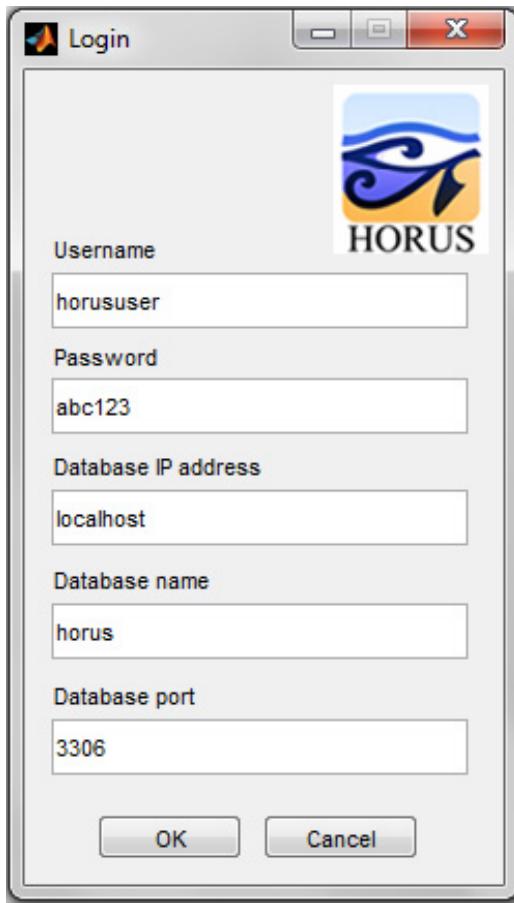


Figura 5.2: Interfaz para ingresar los datos de usuario para el login

el nombre de la base de datos (*dbname*). Cuando un usuario se loguea, debe ingresar toda la información mencionada para la conexión a la base de datos, y se crea el archivo `userinfo.dat`. Cuando, por ejemplo, el usuario cierra una interfaz gráfica, el sistema le pregunta al usuario si desea cerrar la sesión, en caso afirmativo, el archivo `userinfo.dat` es borrado, y en caso negativo no se hace nada. En conclusión, mientras el archivo `userinfo.dat` exista, hay una sesión de HORUS abierta, y se cierra cuando el archivo es borrado.

## 5.1. Setup database

Ver sección 4.2.

## 5.2. Edit database

El sistema HORUS cuenta con una base de datos en la cual se ingresa toda la información generada necesaria para su correcto funcionamiento (ver sección 3.2). Con esta interfaz se puede insertar, actualizar o eliminar la información de una estación, una cámara, un GCP, un sensor o un tipo de medición (los cuales son los parámetros que mide un sensor). Al abrir la interfaz se muestra la opción de insertar una nueva estación a la base de datos de HORUS. Si se desea, con el menú de la parte superior se puede cambiar de opción para actualizar o insertar estaciones, cámaras, GCPs, sensores y tipos de mediciones. A continuación se explica cada una de las opciones de la interfaz.

El comando para ejecutar esta interfaz es: `gui_db_editor`.

### 5.2.1. Estación

En la figura 5.3 se muestra la opción de insertar una nueva estación en la base de datos, y en la figura 5.4 la opción de actualizar o eliminar una estación existente.

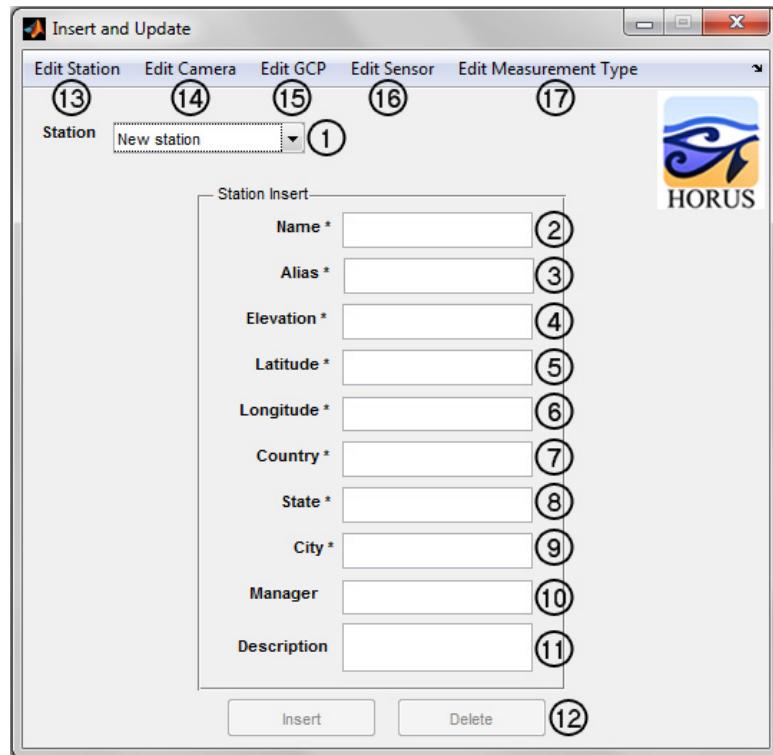


Figura 5.3: Interfaz para insertar una nueva estación

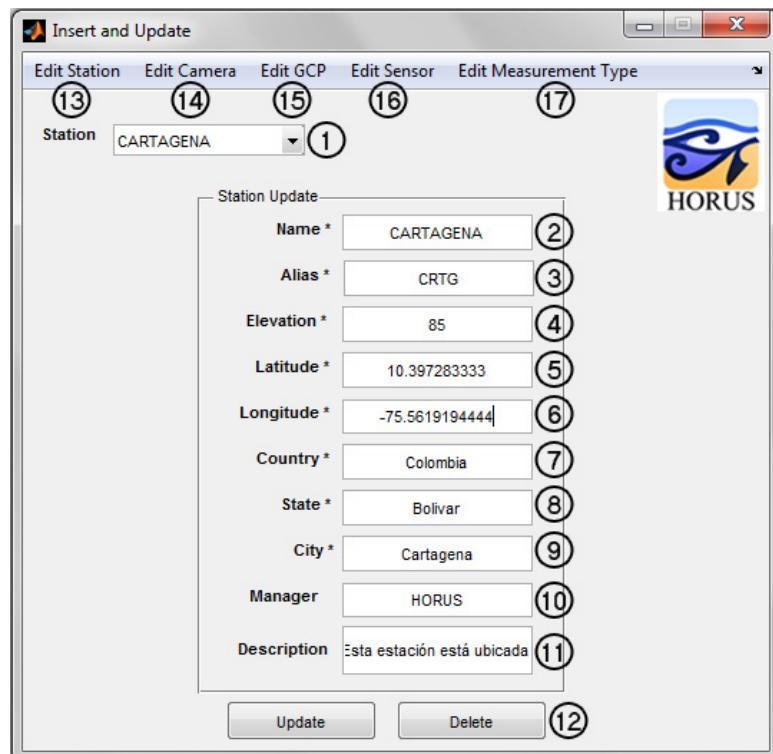


Figura 5.4: Interfaz para actualizar estación

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. En esta lista se debe seleccionar la estación que se desea actualizar o insertar una nueva estación en caso de escoger la opción “*New station*”.
2. Nombre de la estación, éste debe ser diferente al nombre de las demás estaciones y no es posible actualizarlo.
3. Alias de la estación, generalmente, una abreviatura del nombre de la estación. Debe ser de máximo 5 caracteres. Éste debe ser diferente al alias de las demás estaciones y no es posible actualizarlo.
4. Elevación de la estación, éste es un número real.
5. Latitud de la estación, éste es un número real.
6. Longitud de la estación, éste es un número real.
7. País en donde se encuentra la estación.
8. Estado del país donde se encuentra la estación.
9. Ciudad del estado donde se encuentra la estación.
10. Nombre del responsable de la estación (opcional).
11. Descripción general de la estación (opcional).
12. Botones para insertar/actualizar y eliminar el ítem seleccionado. El primer botón se habilita cuando se ha ingresado la información correctamente, y se comprueba que no haya campos vacíos que sean obligatorios y los campos (3), (4) y (5) sean numéricos. El segundo botón se habilita cuando se selecciona la estación en el campo (1). Al terminar, se mostrará una ventana indicando el éxito o el fracaso de la operación.
13. Menú para insertar, actualizar o eliminar una estación.
14. Menú para insertar, actualizar o eliminar una cámara.
15. Menú para insertar, actualizar o eliminar un GCP.
16. Menú para insertar, actualizar o eliminar un sensor.
17. Menú para insertar, actualizar o eliminar un tipo de medición.

En la figura 5.5 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

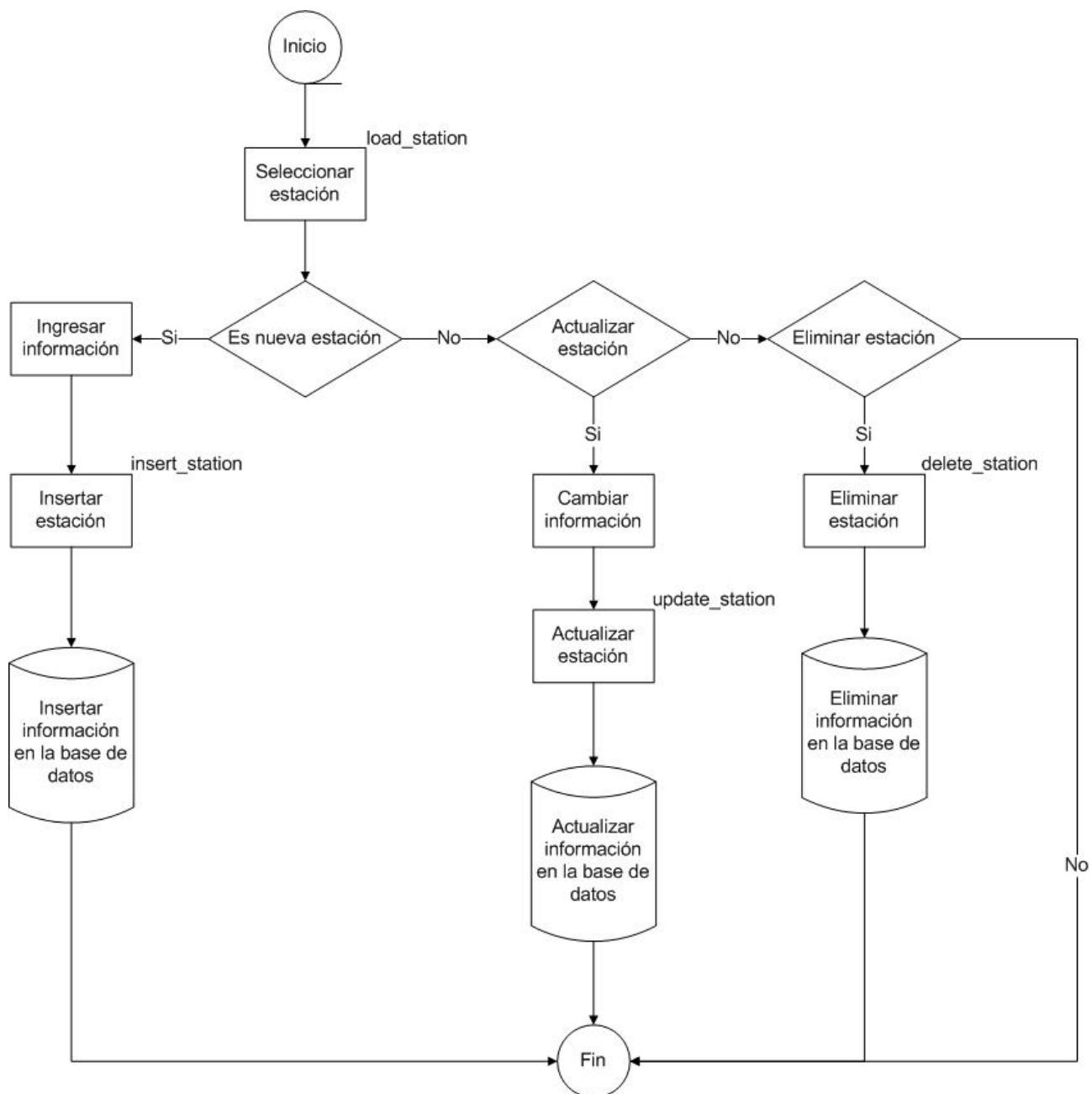


Figura 5.5: Diagrama de flujo interfaz DB Editor - Estación

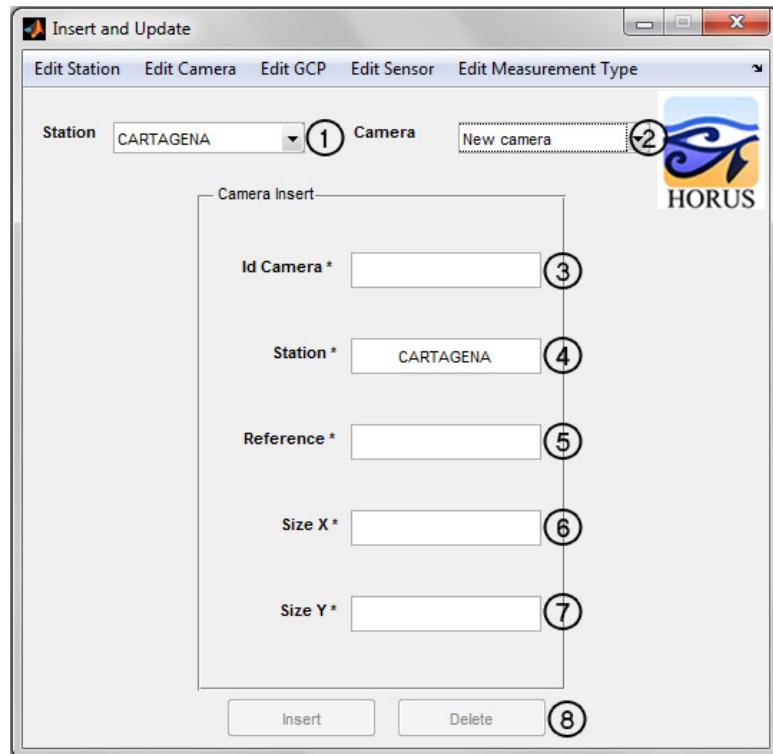


Figura 5.6: Interfaz para insertar una nueva cámara

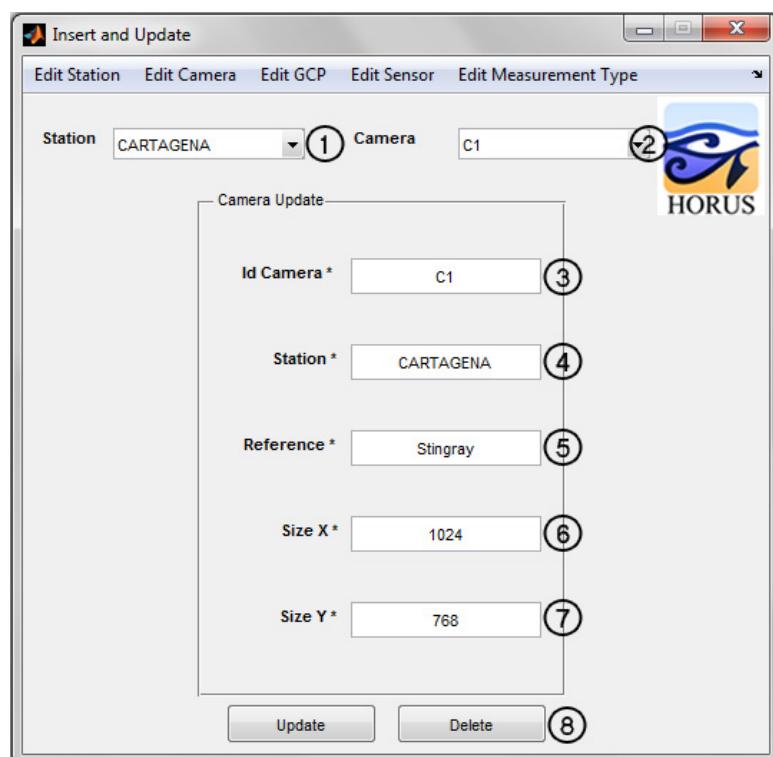


Figura 5.7: Interfaz para actualizar una cámara

### 5.2.2. Cámara

En la figura 5.6 se muestra la opción de insertar una cámara en la base de datos y en la figura 5.7 la opción de actualizar o eliminar.

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. En esta lista se debe seleccionar la estación en la cual se encuentra la cámara a actualizar o en donde se desea insertar una nueva cámara.
2. En esta lista se debe seleccionar la cámara que se desea actualizar o insertar una nueva cámara en caso de escoger la opción “*New camera*”.
3. ID para la cámara, por lo general es 'C' seguido de un número. Éste debe ser único para cada estación y no es posible actualizarlo.
4. En este campo se visualiza la estación asociada a la cámara. Este campo no es posible actualizarlo.
5. Referencia de la cámara.
6. Ancho de la imagen capturada por la cámara. Este valor debe ser numérico y entero.
7. Alto de la imagen capturada por la cámara. Este valor debe ser numérico y entero.
8. Botones para insertar/actualizar y el otro para eliminar una cámara. El primer botón se habilita cuando se ha ingresado la información correctamente, y se comprueba que no haya campos vacíos y que los campos (6) y (7) sean numéricos. El segundo botón se habilita cuando se selecciona la estación en la lista (1) y la cámara en la lista (2). Al terminar, se mostrará una ventana indicando el éxito o el fracaso de la operación.

En la figura 5.8 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

### 5.2.3. GCP

En la figura 5.9 se muestra la opción de insertar un punto de control (GCP) en la base de datos y en la figura 5.10 la opción de actualizar o eliminar.

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. En esta lista se debe seleccionar la estación en la cual se encuentra el GCP a actualizar o insertar.
2. En esta lista se debe seleccionar el GCP que se desea actualizar o insertar un nuevo GCP en caso de escoger la opción “*New GCP*”. Hay otra opción en esta lista para importar GCPs desde un archivo de Excel externo. Al seleccionar esta opción, se abre una nueva ventana para seleccionar un archivo de Excel que debe contener los datos de los GCPs que se desean importar a la base de datos. El archivo de Excel debe tener la estructura mostrada en la figura 5.11. Las columnas contienen en orden, los IDs de los GCPs los cuales deben ser numéricos y únicos para cada estación, los nombres de los GCPs los cuales deben ser únicos para cada estación, los valores de la coordenada ( $X$ ,  $Y$ ,  $Z$ ) del GCP en el sistema coordenado correspondiente, estas tres últimas columnas deben ser

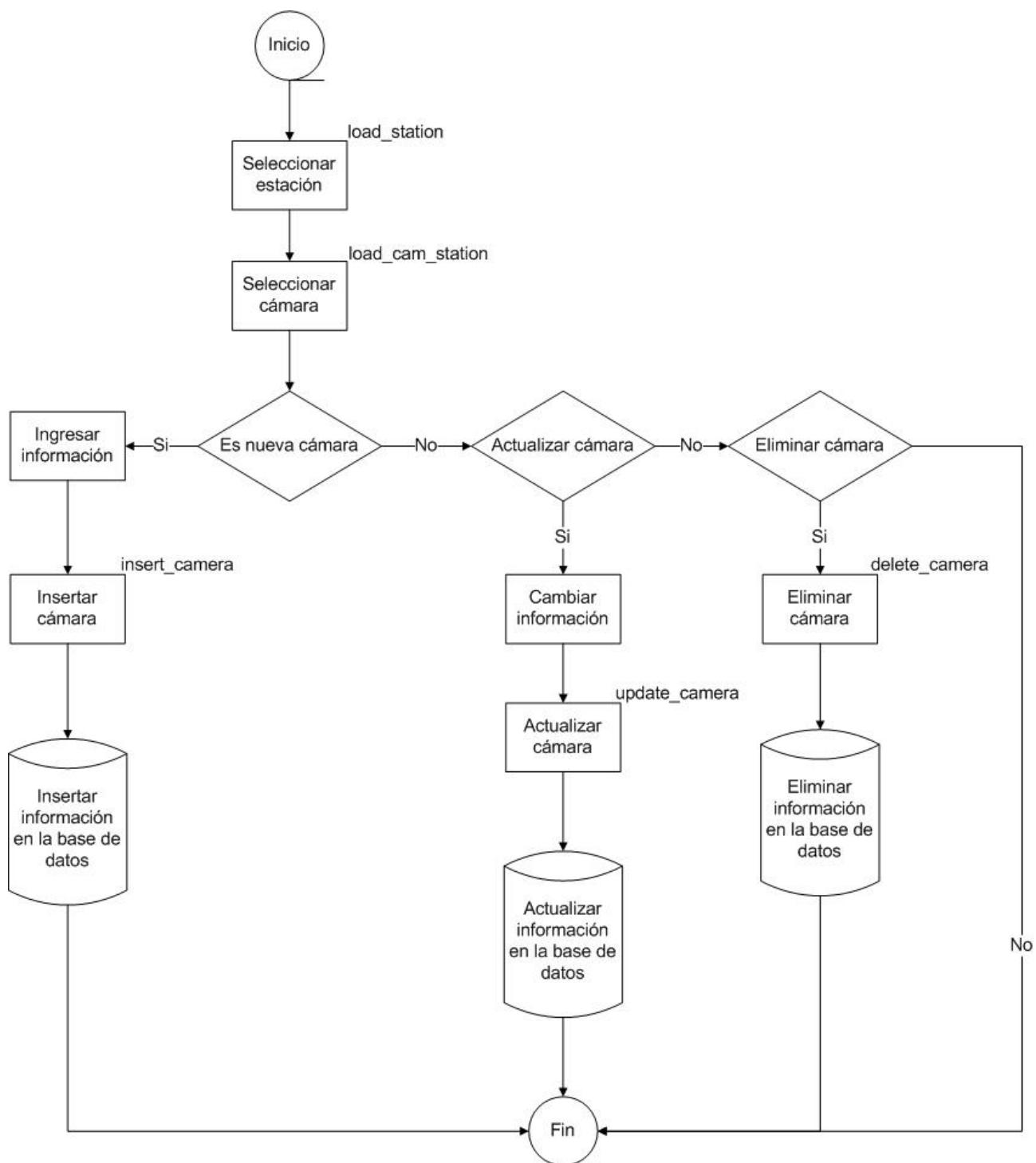


Figura 5.8: Diagrama de flujo interfaz DB Editor - Cámara

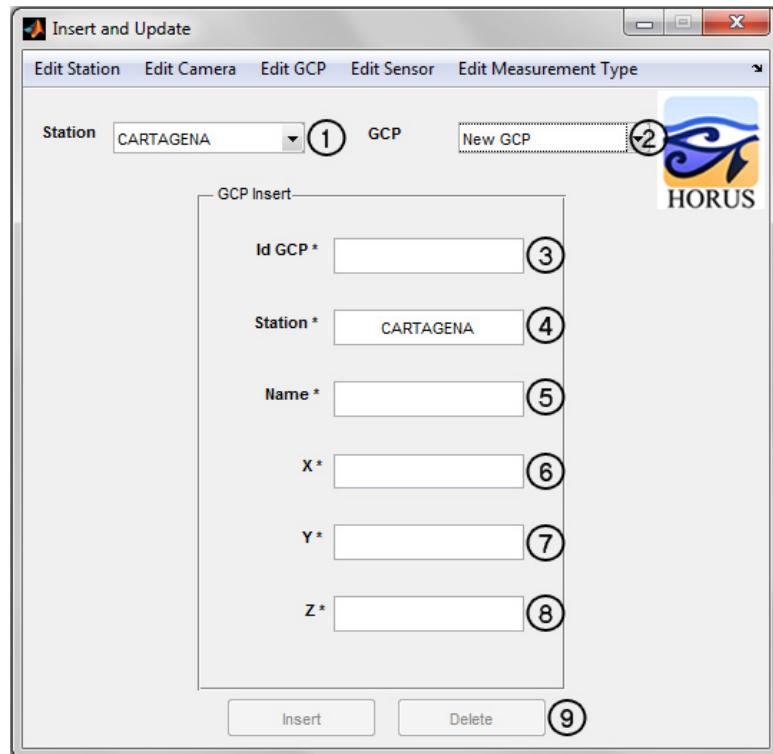


Figura 5.9: Interfaz para insertar un nuevo GCP

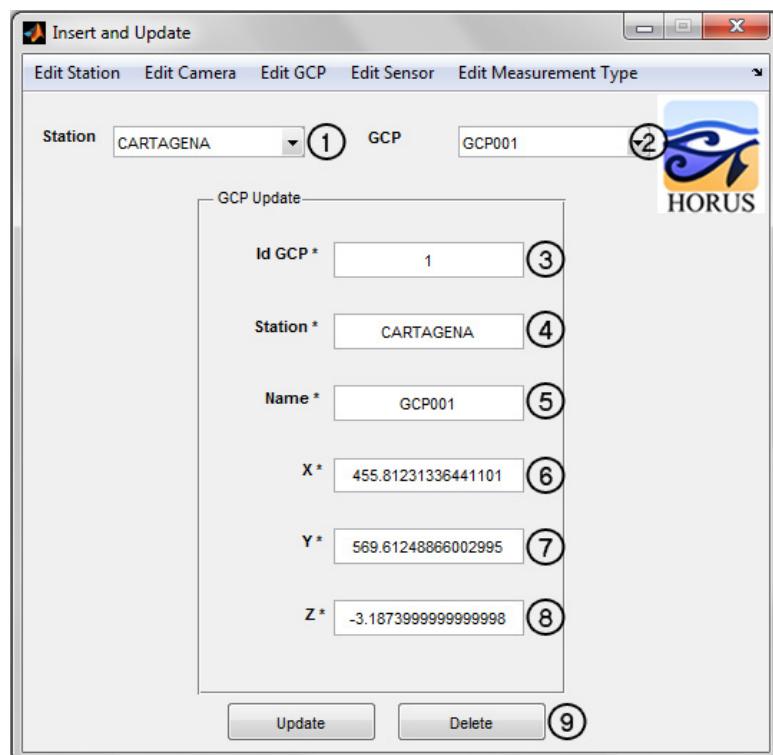


Figura 5.10: Interfaz para actualizar un GCP

	A	B	C	D	E
1		1 GCP001	455.812313	569.612489	-3.1874
2		2 GCP002	287.434999	488.902405	-2.3615
3		3 GCP003	295.005854	491.828358	-2.4437
4		4 GCP004	298.183501	483.830749	-2.3268
5		5 GCP005	333.839192	514.136837	-1.3936
6		6 GCP006	334.799637	515.288255	-2.0715
7		7 GCP007	364.188881	517.130135	-2.3973
8		8 GCP008	368.185622	509.514491	-1.0675
9		9 GCP009	375.776591	512.661295	-2.3551
10		10 GCP010	371.857618	520.423063	-2.3425
11		11 GCP011	410.728673	518.029656	-2.5183
12		12 GCP012	419.309245	522.158967	-2.2715
13		13 GCP013	441.92835	532.851742	-2.4167
14		14 GCP014	460.031885	547.945683	-1.8381
15		15 GCP015	456.371051	555.632102	-2.0626

Figura 5.11: Estructura de excel para importar GCPs

numéricas. El archivo de Excel sólo debe tener una hoja de cálculo, todas las demás deben ser eliminadas, también el separador decimal del archivo debe ser un punto (.).

Este archivo no debe tener ningún encabezado, la primera fila debe contener los datos del primer GCP a insertar en la base de datos. Después de seleccionar el archivo se debe dar clic en el botón insertar. Al terminar el proceso saldrá un mensaje indicando la cantidad de GCPs insertados exitosamente. Se debe tener en cuenta que los GCPs importados no reemplazarán los existentes, si se desean actualizar se deberán eliminar todos y luego importarlos o actualizar cada uno a la vez.

En las opciones, también se puede seleccionar o eliminar todos los GCPs (incluyendo las marcaciones del usuario) en la estación que se seleccionó anteriormente. Después de seleccionar esta opción se debe dar clic en el botón “*Delete*”, al terminar el proceso saldrá un mensaje indicando el éxito o el fracaso de la operación.

3. ID para el GCP, es un entero que debe ser único para cada estación. Este campo no es posible actualizarlo.
4. Estación asociada al GCP. Este campo no es posible actualizarlo.
5. Nombre del GCP, por lo general es GCP seguido por tres dígitos (e.g. GCP001). Este campo debe ser único para cada estación.
6. Valor real de la coordenada *x* del GCP.
7. Valor real de la coordenada *y* del GCP.
8. Valor real de la coordenada *z* del GCP.
9. Botones para insertar/actualizar y eliminar un GCP. El primer botón se habilita cuando se ha ingresado la información correctamente, y se comprueba que no haya campos vacíos y los campos (6), (7) y (8) sean numéricos. El segundo botón se habilita cuando se selecciona la estación en la lista (1) y el GCP en la lista (2). Al terminar se mostrará una ventana indicando el éxito o el fracaso de la operación.

En la figura 5.12 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

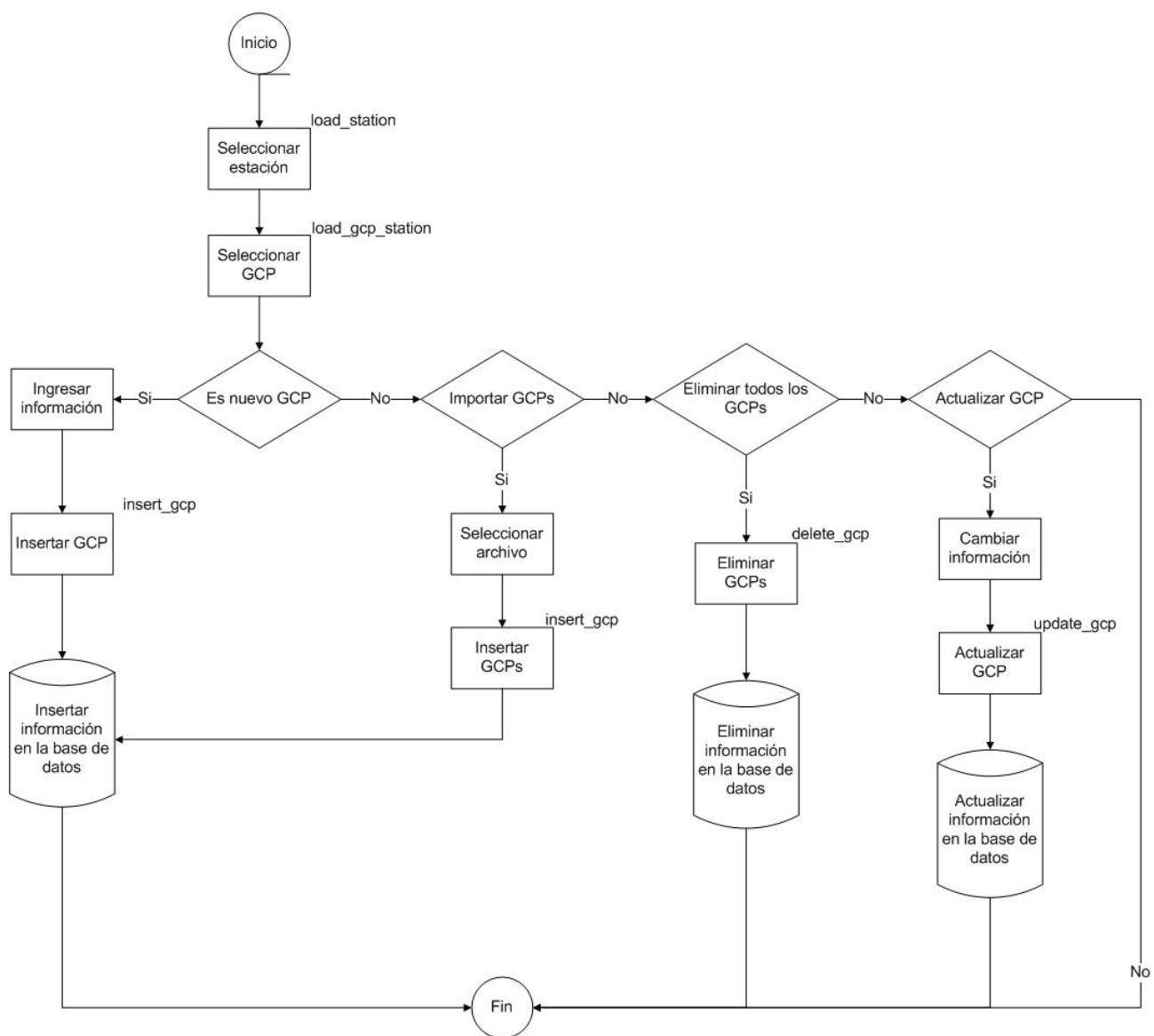


Figura 5.12: Diagrama de flujo interfaz DB Editor - GCP

#### 5.2.4. Sensor

Los sensores son los equipos utilizados para hacer mediciones, por ejemplo, una boyas puede medir altura de ola y período de ola. En la figura 5.13 se muestra la opción de insertar un sensor en la base de datos y en la figura 5.14 la opción de actualizar o eliminar.

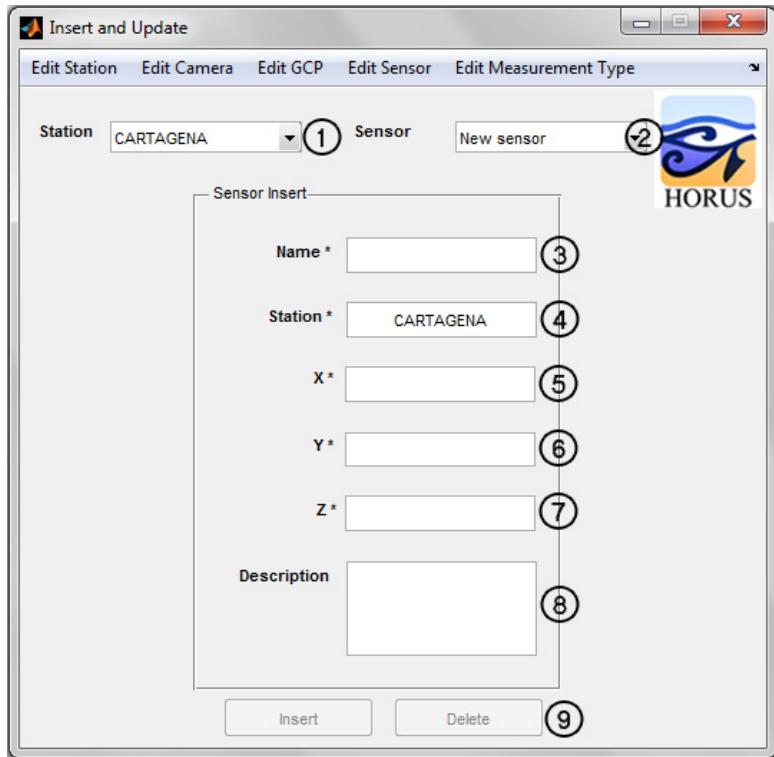


Figura 5.13: Interfaz para insertar un nuevo sensor

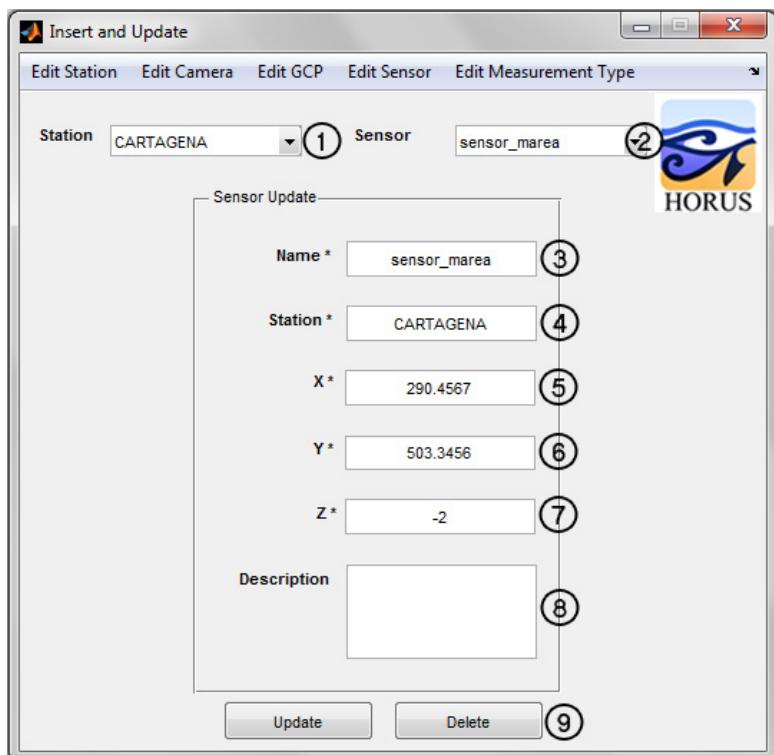


Figura 5.14: Interfaz para actualizar un sensor

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. En esta lista se debe seleccionar la estación en la cual se encuentra el sensor a actualizar o en donde se desea insertar el sensor.
2. En esta lista se debe seleccionar el sensor que se desea actualizar o insertar un nuevo sensor en caso de escoger la opción “*New sensor*”.
3. Nombre para el sensor el cual debe ser único para cada estación. Este campo no es posible actualizarlo.
4. Estación asociada al sensor. Este campo no es posible actualizarlo.
5. Valor real de la coordenada *x* del sensor.
6. Valor real de la coordenada *y* del sensor.
7. Valor real de la coordenada *z* del sensor.
8. Descripción verbal del sensor.
9. Botones para insertar/actualizar y eliminar un sensor. El primer botón se habilita cuando se ha ingresado la información correctamente, se comprueba que sólo el campo de descripción pueda estar vacío y los campos (5), (6) y (7) sean numéricos. El segundo botón se habilita cuando se selecciona la estación en la lista (1) y el sensor en la lista (2). Al terminar, se mostrará una ventana indicando el éxito o el fracaso de la operación.

En la figura 5.15 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

### 5.2.5. Tipo de Medición

Los tipos de medición son los parámetros que pueden ser medidos por un sensor, por ejemplo, marea, altura significante de ola, período pico del oleaje, entre otros. Un tipo de medición siempre depende de un sensor, si no se tiene la información exacta del sensor se podrán crear sensores artificiales. Pueden haber varios tipos de mediciones asociadas a un sensor.

En la figura 5.16 se muestra la opción de insertar un tipo de medición en la base de datos y en la figura 5.17 la opción de actualizar o eliminar.

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. En esta lista se debe seleccionar la estación en la cual se encuentra el tipo de medición a actualizar o en donde se desea insertar el tipo de medición.
2. En esta lista se debe seleccionar el tipo de medición que se desea actualizar o insertar un nuevo tipo de medición en caso de escoger la opción “*New measurement type*”.
3. En esta lista se debe seleccionar el sensor con el cual se hicieron las mediciones.
4. Nombre del parámetro medido, el cual debe ser único para cada estación y de preferencia que sea una abreviatura. Este campo no es posible actualizarlo.
5. Lista con los tipos de dato que puede ser “time series” si los datos son de una serie de tiempo, o “matrix” si los datos son en tres dimensiones.

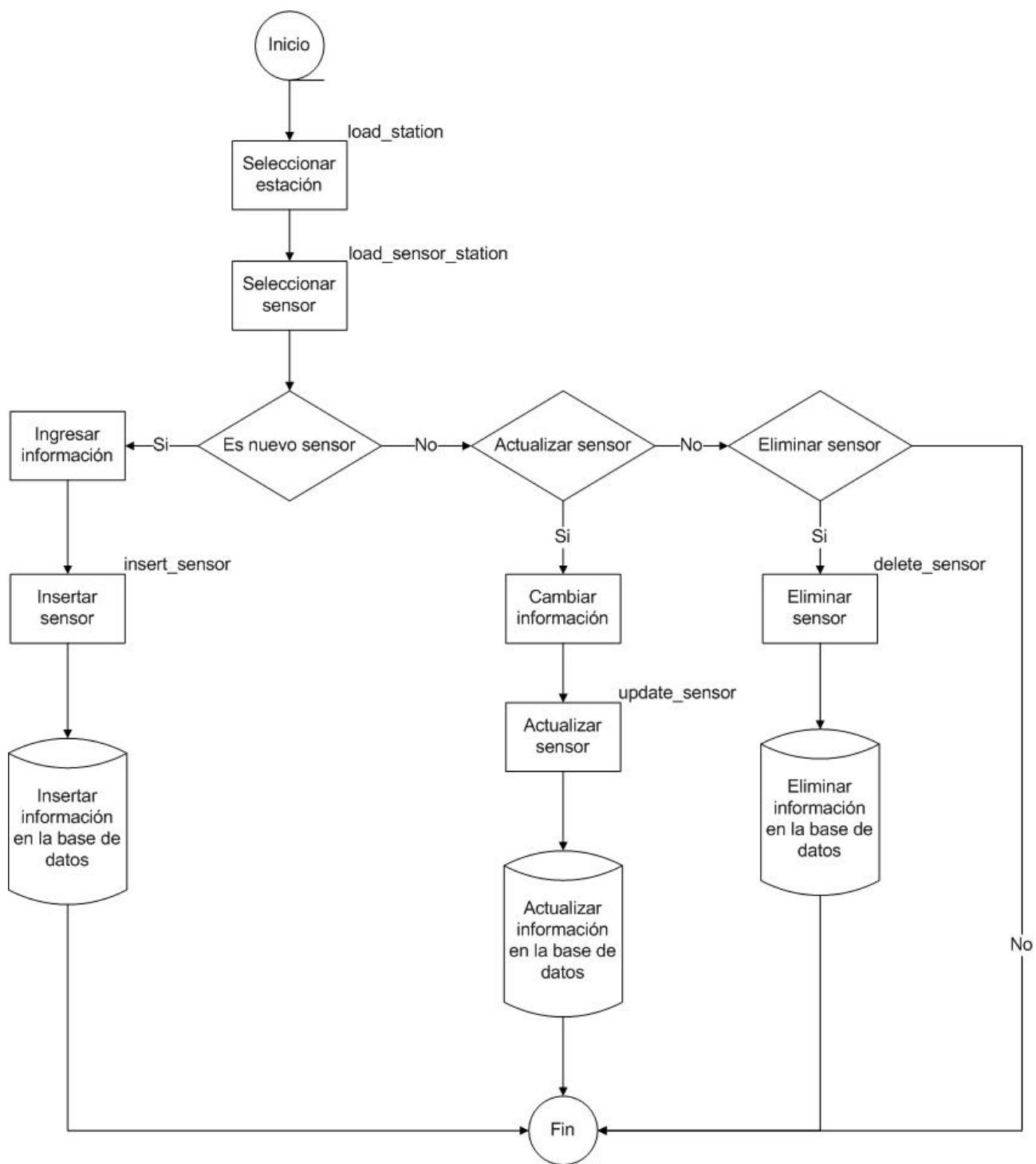


Figura 5.15: Diagrama de flujo interfaz DB Editor - Sensor

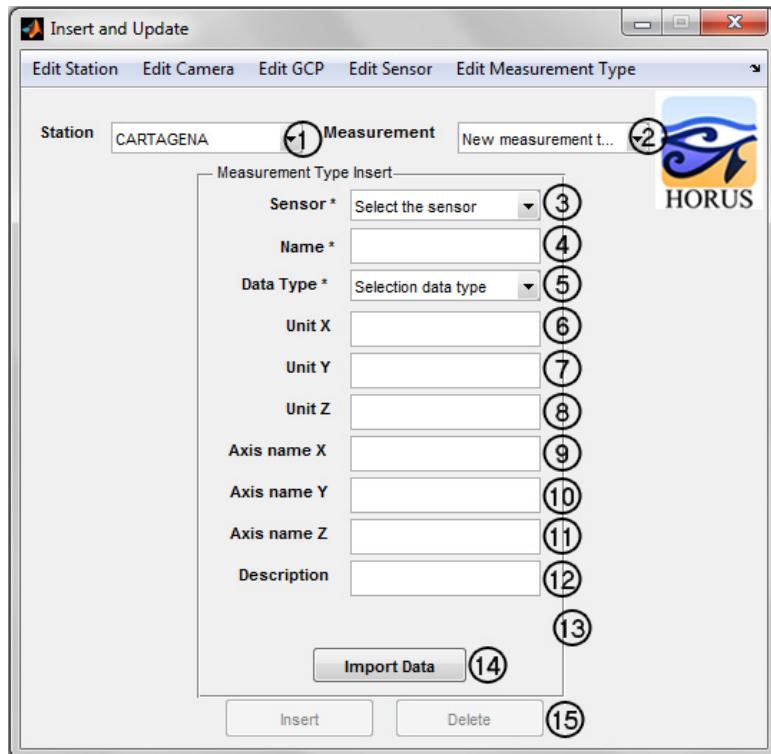


Figura 5.16: Interfaz para insertar un nuevo tipo de medición

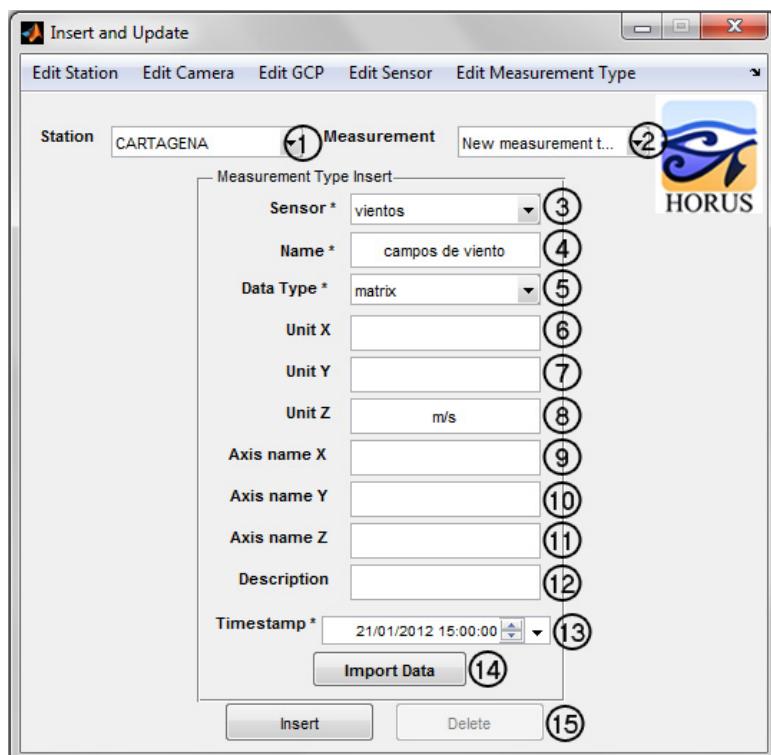


Figura 5.17: Interfaz para actualizar un tipo de medición

6. Abreviatura de la unidad de medida del eje horizontal ( $x$ ). Si es una serie de tiempo sería la unidad de tiempo (horas, minutos).
7. Abreviatura de la unidad de medida del eje vertical ( $y$ ).
8. Abreviatura de la unidad de medida del eje  $z$ . Si es una serie de tiempo este campo se desactiva.
9. Etiqueta del eje  $x$  en la gráfica. Este campo no es necesario si es una serie de tiempo.
10. Etiqueta del eje  $y$  en la gráfica. Este campo no es necesario si es una serie de tiempo.
11. Título de la gráfica si se trata de un tipo de dato matriz, en caso contrario se desactiva.
12. Descripción del tipo de medición.
13. En caso de ser datos de tipo matriz se mostrará un calendario en donde se escogerá la fecha de la medición. Éste puede tomar cualquier fecha menor al instante en que se esté haciendo la inserción.
14. Botón para agregar los datos de la medición a la base de datos, el cual es obligatorio sólo si se desea agregar información. Haciendo clic en este botón se abrirá una ventana en la cual se escogerá el archivo que tiene los datos de la medición. Este archivo puede tener formato MAT o Excel, y no debe tener ningún encabezado. La estructura que debe tener el archivo es la siguiente:

Si el tipo de dato es serie de tiempo, en la primera columna debe estar la fecha de la medición en el formato que utiliza la función `datenum` de MATLAB. En la segunda columna debe estar el valor de la medición para esa fecha. En la figura 5.18 se muestra la estructura necesaria.

H5 <4737x2 double>		
	1	2
1	7.3393e+05	0.4774
2	7.3393e+05	0.4808
3	7.3393e+05	0.4843
4	7.3393e+05	0.4965
5	7.3393e+05	0.5205
6	7.3393e+05	0.5519
7	7.3393e+05	0.5726
8	7.3393e+05	0.6217
9	7.3393e+05	0.6819
10	7.3393e+05	0.7537
11	7.3393e+05	0.8268
12	7.3393e+05	0.8757
13	7.3393e+05	0.8955
14	7.3393e+05	0.8804
15	7.3393e+05	0.8492

Figura 5.18: Estructura de MATLAB si el tipo de medición es serie

Si el tipo de dato es matriz en la primera columna se debe poner el valor de  $x$  en la segunda columna el valor de  $y$  y en la tercera el valor de  $z$ . En la figura 5.19 se muestra la estructura necesaria.

	A	B	C
1	0.0311	0	0.8147
2	0.0311	24	0.9058
3	0.0311	48	0.1270
4	0.0311	72	0.9134
5	0.0311	96	0.6324
6	0.0311	120	0.0975
7	0.0311	144	0.2785
8	0.0311	168	0.5462
9	0.0311	192	0.9575
10	0.0311	216	0.9648
11	0.0311	240	0.1576
12	0.0311	264	0.9705
13	0.0311	288	0.9571
14	0.0311	312	0.4853
15	0.0311	336	0.8002

Figura 5.19: Estructura de Excel si el tipo de medición es matriz

Si el archivo tiene formato MAT, éste sólo podrá contener una matriz que tenga los datos, si el archivo es de Excel sólo debe tener una hoja de cálculo, todas las demás deben ser eliminadas, también el separador decimal del archivo debe ser un punto (.).

15. Botones para insertar/actualizar y eliminar un tipo de medición. El primer botón se habilita cuando se ha ingresado la información correctamente, y se comprueba que los campos con \* no estén vacíos. El segundo botón se habilita cuando se selecciona la estación en la lista (1) y el tipo de medición en la lista (2). Al terminar se mostrará una ventana indicando el éxito o el fracaso de la operación.

Si se ha insertado un tipo de medición y luego se quiere agregar más datos a éste se puede hacer seleccionándolo en el campo (2) y luego seleccionando el archivo por medio del campo (11), y finalmente dando clic en el botón “*Update*”.

En la figura 5.20 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

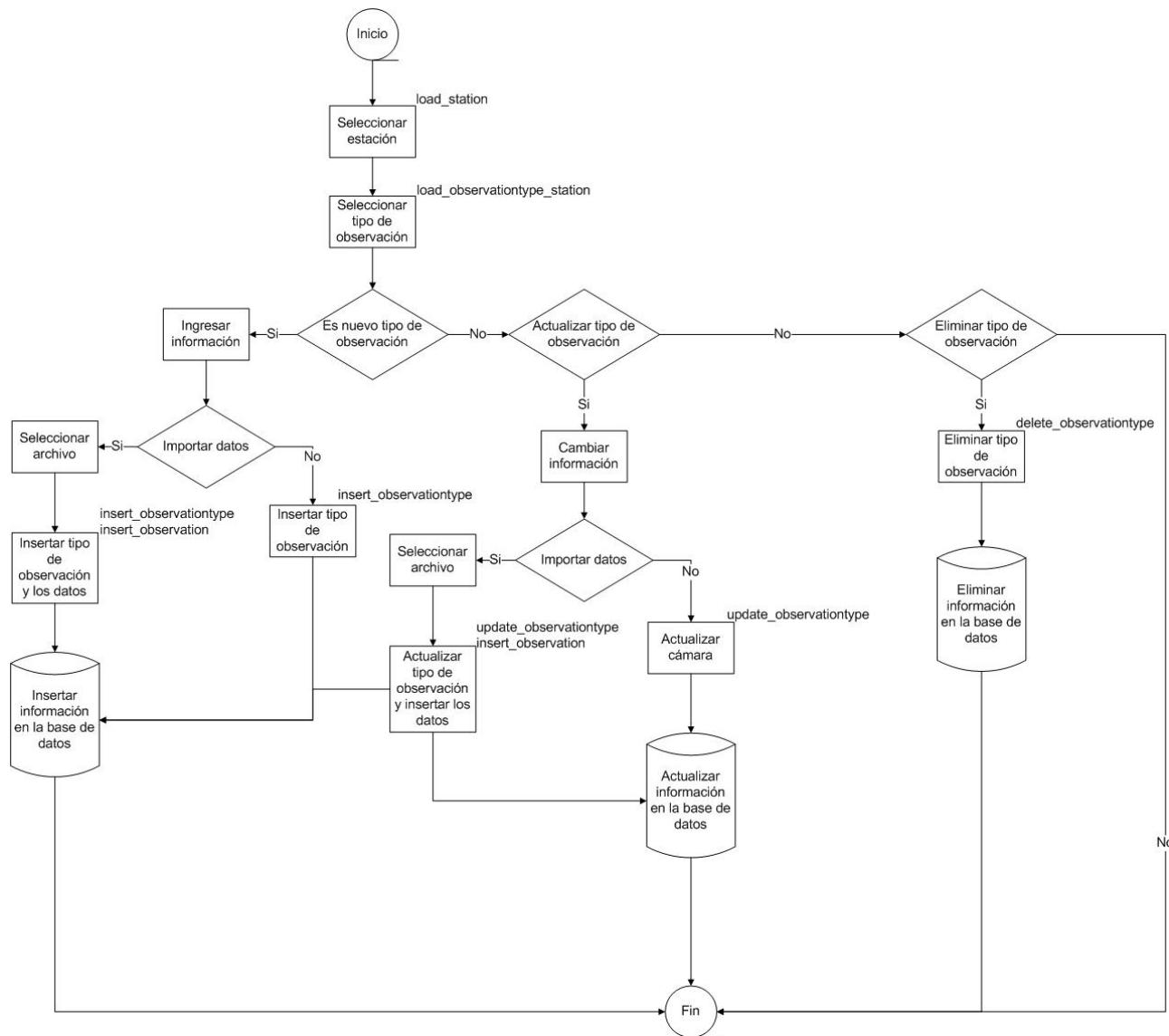


Figura 5.20: Diagrama de flujo interfaz DB Editor - Tipo de Medición

### 5.2.6. Generar rutas de imágenes

Todas las imágenes generadas por HORUS se deben almacenar en el disco duro local. La interfaz *Generate Image Paths*, permite, dado un directorio raíz, generar automáticamente las rutas para los diferentes tipos de imágenes procesadas (oblicuas, rectificadas, oblicuas–fusionadas, rectificadas–fusionadas y miniaturas).

En la figura 5.21 se muestra la interfaz para generar las rutas de las imágenes automáticamente.

Figura 5.21: Interfaz para generar las rutas automáticamente

Las componentes numeradas en la figura se explican a continuación:

1. Lista donde el usuario puede seleccionar la estación para la cual, las imágenes van a estar asociadas.
2. Selección de la ruta raíz para las imágenes oblicuas, capturadas desde la estación de captura.
3. Selección de la ruta raíz para las imágenes procesadas.
4. Botón para guardar la configuración en el archivo XML `path_info.xml` del directorio de HORUS data.

### 5.3. Create new user

Si se quiere crear un nuevo usuario con permisos limitados (e.g. un usuario de sólo lectura), se ejecuta la interfaz gráfica `gui_create_user` (ver figura 5.22) donde se especifica el nombre de usuario, la contraseña, la base de datos a la que tiene permiso de conectarse y si tiene permiso de consultar (*query*), insertar (*insert*), borrar (*delete*) o actualizar (*update*) la información en la base de datos.



Figura 5.22: Interfaz para crear un nuevo usuario con permisos limitados

## 5.4. Configure cameras

En esta sección se describe la interfaz utilizada para configurar las cámaras conectadas al computador de captura con los parámetros de captura requeridos. Para utilizar esta interfaz es necesario disponer del toolbox de Adquisición de Imágenes de MATLAB.

Para configurar las cámaras se debe tener instalado el adaptador correspondiente en el caso de cámaras *Firewire* o cámaras web. Si se trata de una cámara web, por lo general, el sistema operativo reconoce el controlador, sino, hay que instalarlo manualmente usando el instalador que provee la marca de la cámara. Por ejemplo, si se trata de cámaras *Firewire* AVT (*Allied Vision Technologies*) se debe instalar el software *FirePackage* disponible en la página web <http://www.alliedvisiontec.com/emea/products/software/windows/avt-firepackage.html>. Para otro tipo de cámaras *Firewire* se debe instalar el controlador universal CMU disponible en <http://www.cs.cmu.edu/~iwan/1394/>.

Luego de instalar el controlador en el computador, para utilizar las cámaras en MATLAB es necesario registrar el adaptador correspondiente con el comando `imaqregister`. MATLAB dispone de un adaptador por defecto para las cámaras web; en el sistema operativo Windows, este adaptador es `winvideo`; en Linux, es `linuxvideo` (*Nota:* Otro adaptador común que viene incluído en MATLAB es `coreco`, para cámaras *DALSA-CORECO*). Si éste es el caso, no es necesario registrar un adaptador, pero para otro tipo de cámaras como las *Firewire*, sí es necesario registrarlos (si está soportado por la marca correspondiente). Por ejemplo, si el adaptador es `avtmatlabadaptor.dll` (el correspondiente a las cámaras *Firewire* de la marca AVT) y se encuentra en el directorio C:, el comando para registrarlo en MATLAB sería:

```
> imaqregister('C:\avtmatlabadaptor.dll')  
> imaqreset
```

En la figura 5.23 se muestra la interfaz de configuración de cámaras.

El comando para ejecutar esta interfaz es: `gui_configure_cameras`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista donde se puede escoger el adaptador.
2. Lista para escoger el dispositivo conectado al computador de captura.
3. Lista para escoger el formato del dispositivo. Se compone de dos partes, el formato de píxeles (e.g. RGB, YUV) y el tamaño de la imagen.
4. Lista para seleccionar la estación de captura, previamente almacenada y configurada en la base de datos.
5. Lista para asociar una cámara previamente configurada en la base de datos, con el dispositivo físico seleccionado actualmente.
6. Lista de *framerates* disponibles para el dispositivo.
7. Modo de ganancia del dispositivo.
8. Modo de disparo del dispositivo.
9. Modo de balance de blancos del dispositivo.

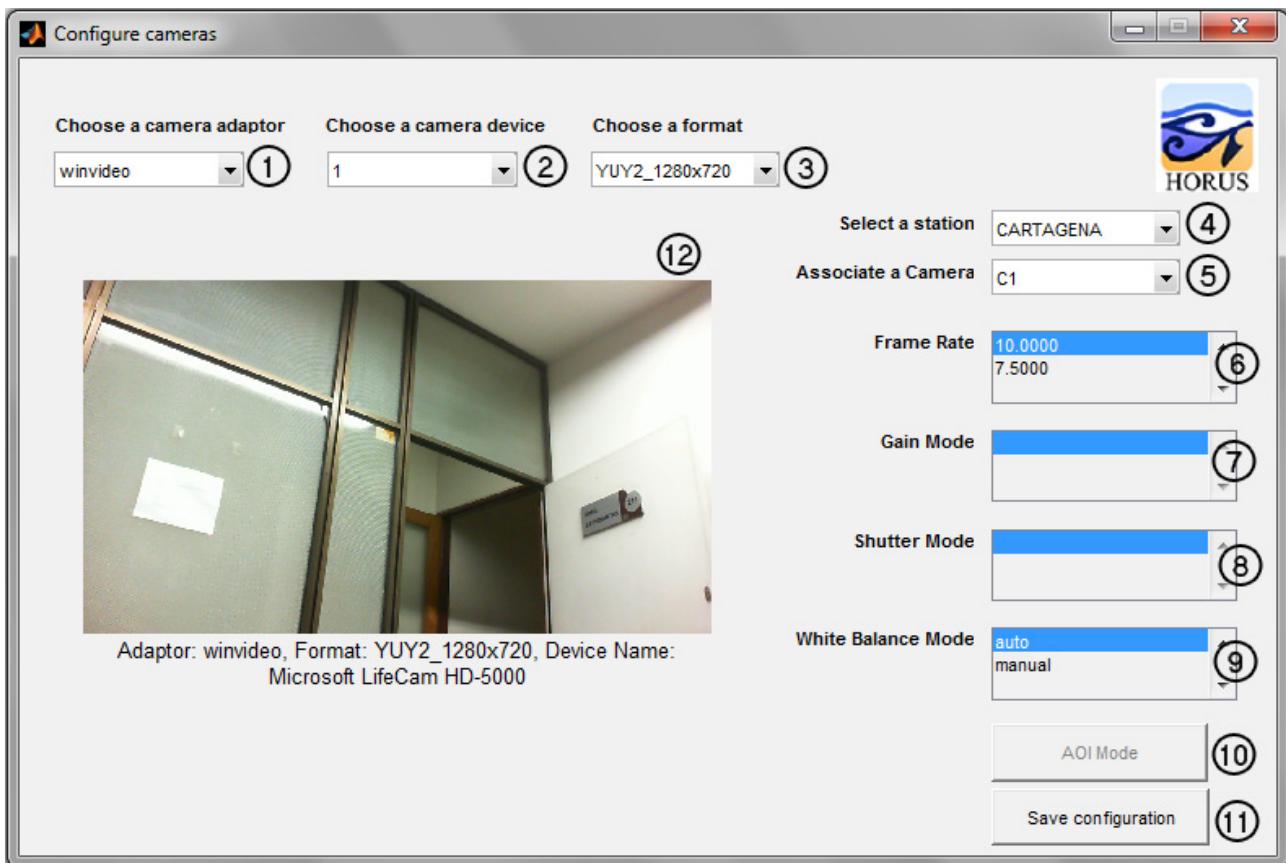


Figura 5.23: Interfaz de Configuración de cámaras

10. Si la cámara es AVT *Firewire*, se puede seleccionar un área de interés (AOI) la cual es la región de la imagen que será utilizada para calcular los valores de la Ganancia, Disparo y Balance de blancos en caso de ser escogidos en la configuración de la cámara.
11. Botón para guardar la configuración de la cámara. Esta configuración se guarda en un archivo XML llamado `capture_info.xml`.
12. Previsualización de la imagen de la cámara.

En la figura 5.24 se muestra el diagrama de flujo para la configuración de las cámaras, donde para cada proceso se nombra la función que utiliza.

## 5.5. Setup image paths

Actualmente, el sistema HORUS trabaja con cuatro tipos de imágenes: oblicuas, rectificadas, oblicuas-fusionadas y rectificadas-fusionadas, pero se pueden agregar más a la base de datos cuando sea necesario. De cada uno de estos tipos se pueden generar imágenes miniaturas.

En la sección 5.2.6 se vio cómo se pueden generar automáticamente las rutas para las imágenes. En esta sección se describe la interfaz que sirve para configurar manualmente las rutas de almacenamiento en disco de cada uno de estos tipos de imágenes. En total, son ocho rutas que se deben configurar.

El Toolbox de HORUS cuenta con un archivo XML que contiene las rutas a las imágenes de cada estación, `path_info.xml`. Con esta interfaz se pueden editar estas rutas. En las figuras 5.25 y 5.26 se muestra la interfaz.

El comando para ejecutar esta interfaz es: `gui_paths_editor`.

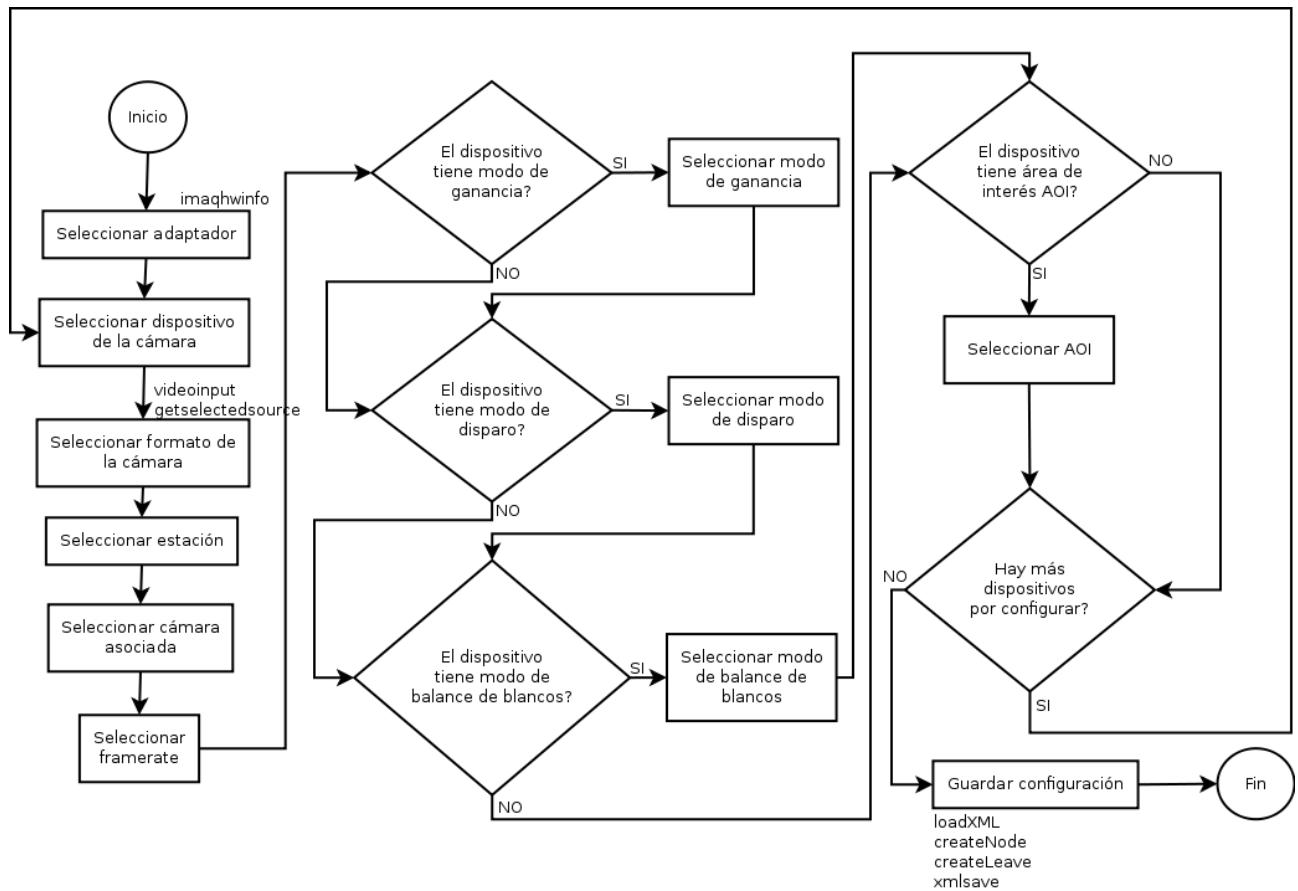


Figura 5.24: Diagrama de flujo para el proceso de configuración de las cámaras

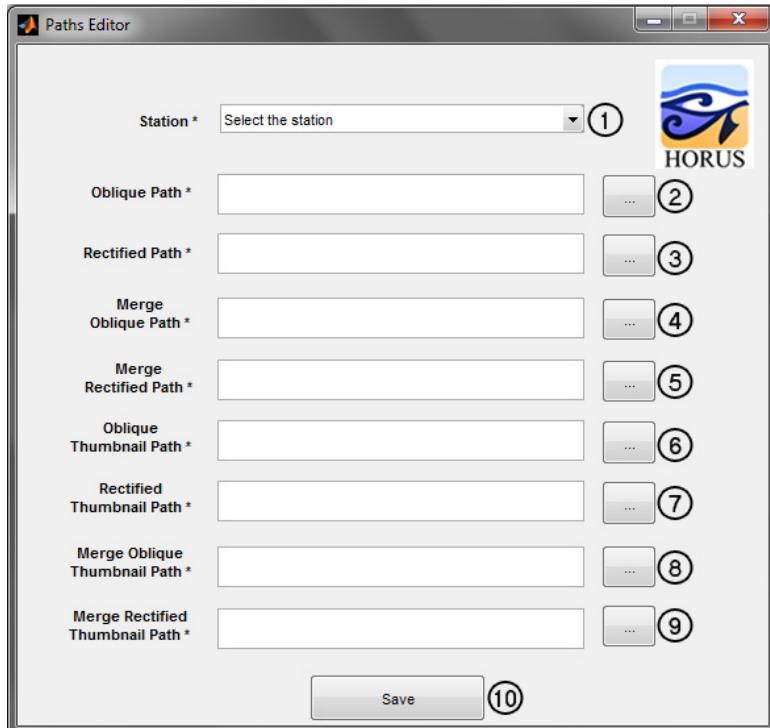


Figura 5.25: Interfaz inicial para editar las rutas

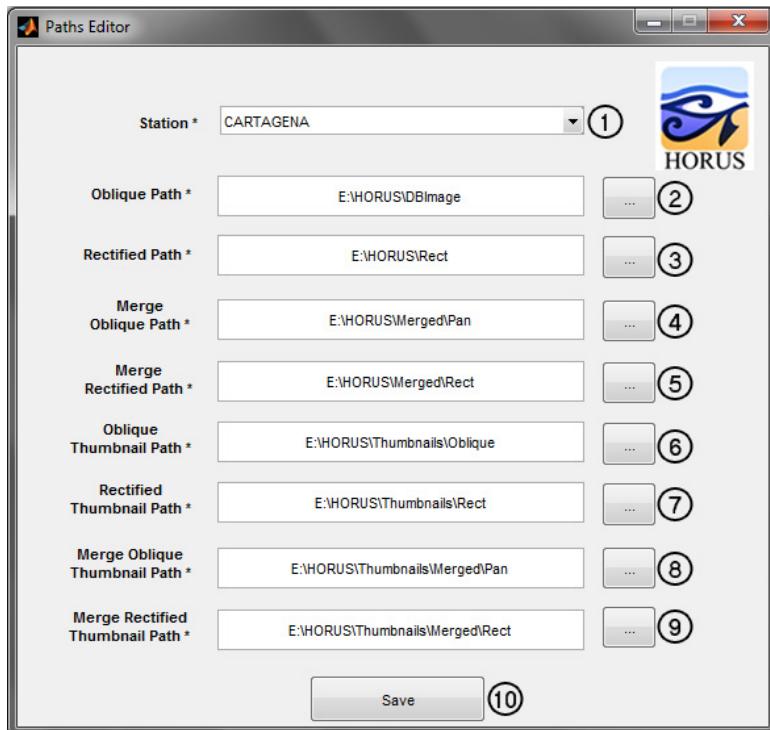


Figura 5.26: Interfaz para editar las rutas con los campos rellenos

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. Se debe seleccionar la estación a la cual se le desea cambiar la ruta de las imágenes. Después de haber seleccionado la estación, se muestran todas las rutas actuales siempre y cuando hayan sido configuradas previamente.

En las opciones de la (2) a la (9) se selecciona la nueva ruta ya sea reescribiéndola en el campo de texto o haciendo clic en el botón “...” que abrirá una nueva ventana en donde se puede seleccionar la nueva ruta. La ruta debe ser a un directorio existente o una dirección web con la estructura <http://IP/ruta> o <http://dominio/ruta>. Se debe tener en cuenta que las rutas que sean direcciones web sólo sirven para hacer la lectura de las imágenes, para las imágenes almacenadas *siempre* se deben guardar en el disco duro.

2. Ruta de las imágenes oblicuas.
3. Ruta de las imágenes rectificadas.
4. Ruta de las imágenes oblicuas–fusionadas.
5. Ruta de las imágenes rectificadas–fusionadas.
6. Ruta de las imágenes oblicuas miniaturas.
7. Ruta de las imágenes rectificadas miniaturas.
8. Ruta de las imágenes oblicuas–fusionadas miniaturas.
9. Ruta de las imágenes rectificadas–fusionadas miniaturas.

- Botón para guardar la configuración en el archivo `path_info.xml`. Al presionar este botón se comprobará que no existan campos vacíos, y que las rutas sean directorios que existan en el disco duro o sean direcciones web, con la estructura `http://IP/ruta` o `http://dominio/ruta`. Si al hacer la comprobación se encuentra que algún campo no cumple con esto, lo mostrará en rojo, el cual deberá cambiarse para poder guardar las rutas. Al terminar se mostrará una ventana indicando el éxito o el fracaso de la operación.

En la figura 5.27 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

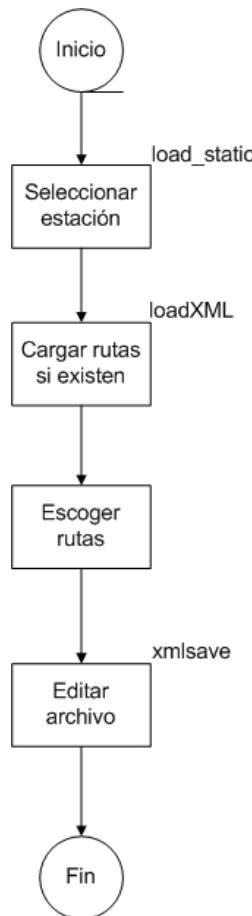


Figura 5.27: Diagrama de flujo interfaz Path Editor

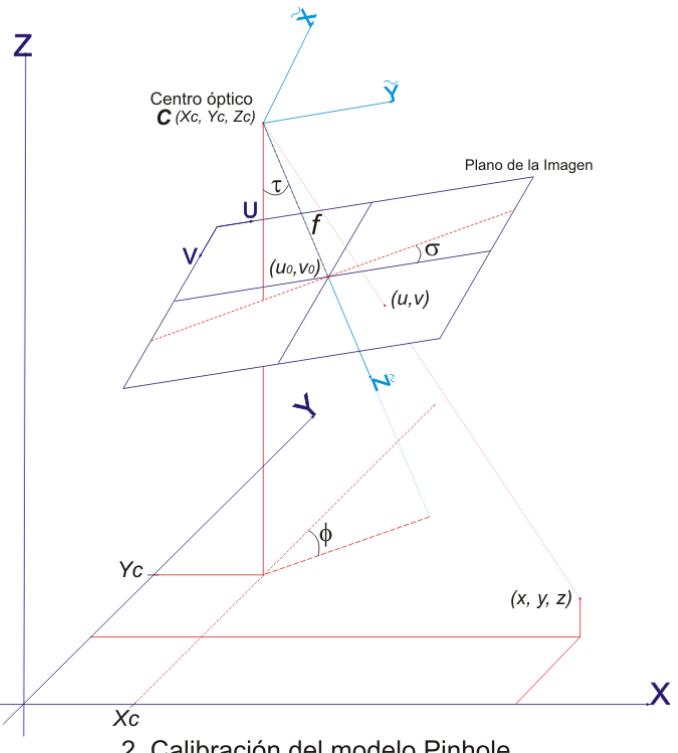
## 5.6. Configure rectification

En HORUS es posible generar calibraciones para la rectificación. En la figura 5.28 se muestra el proceso de calibración. Primero, se eligen unos GCPs y se marca su ubicación en la imagen; estos puntos sirven como referencia para convertir de coordenadas  $(u, v)$  a  $(x, y)$  que finalmente son las que se muestran en la imagen rectificada. Segundo, se calibra el modelo de acuerdo a estos GCPs mediante uno de los tres métodos: Transformada Lineal Directa (*DLT*) [1, 2, 3, 4], *RANSAC-DLT* [7] o *Pinhole* [2, 5, 6, 3] (tiene en cuenta otros factores como la distorsión debida a los lentes de la cámara). Tercero, se elige un ROI que es la región de la imagen que se rectifica. Por último, se rectifica una imagen como medio de validación visual de que el modelo calibrado es correcto.

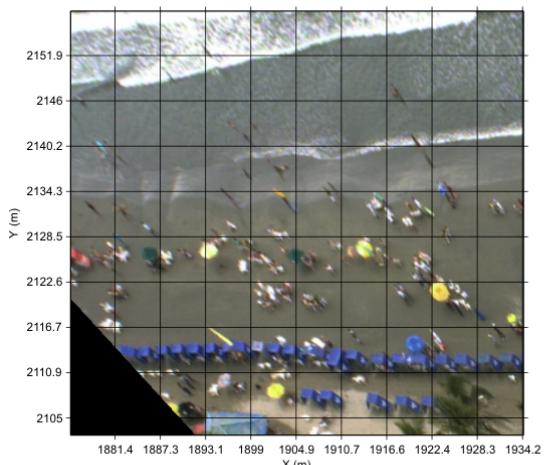
El proceso de calibración para rectificación es fundamental en el sistema HORUS, ya que esto es lo que permite que las imágenes tomadas por las cámaras puedan proveer información útil



1. Elección de puntos de control



2. Calibración del modelo Pinhole



4. Corrección de la perspectiva en la  
región de interés



3. Elección de una región de interés

Figura 5.28: Proceso de calibración

medible para estudiar diferentes fenómenos. Antes de rectificar imágenes es necesario generar las calibraciones de todas las cámaras. Para este proceso se debe contar con un conjunto de GCPs que puedan ser identificados en las imágenes como coordenadas  $(u, v)$ . Estos puntos se marcan en las imágenes correspondientes a las cámaras para las que queremos calcular la calibración, y sirven como insumo para los métodos de optimización.

La interfaz de calibración tiene como fin proveer al usuario un medio para generar una calibración para una cámara en una estación y un tiempo dados. Cada estación debe tener un conjunto de GCPs georreferenciados. El usuario debe escoger un subconjunto entre estos GCPs y marcarlos en la imagen (o cargarlos desde un archivo de EXCEL o MAT) y puede escoger varios métodos de calibración, entre ellos, *DLT*, *RANSAC-DLT* y *Pinhole*. Si el método elegido es el *Pinhole* se puede alimentar el método con estimados iniciales, los cuales corresponden a los parámetros del último modelo guardado en la base de datos o los generados por el método *DLT*. En la figura 5.29 se muestra la interfaz.

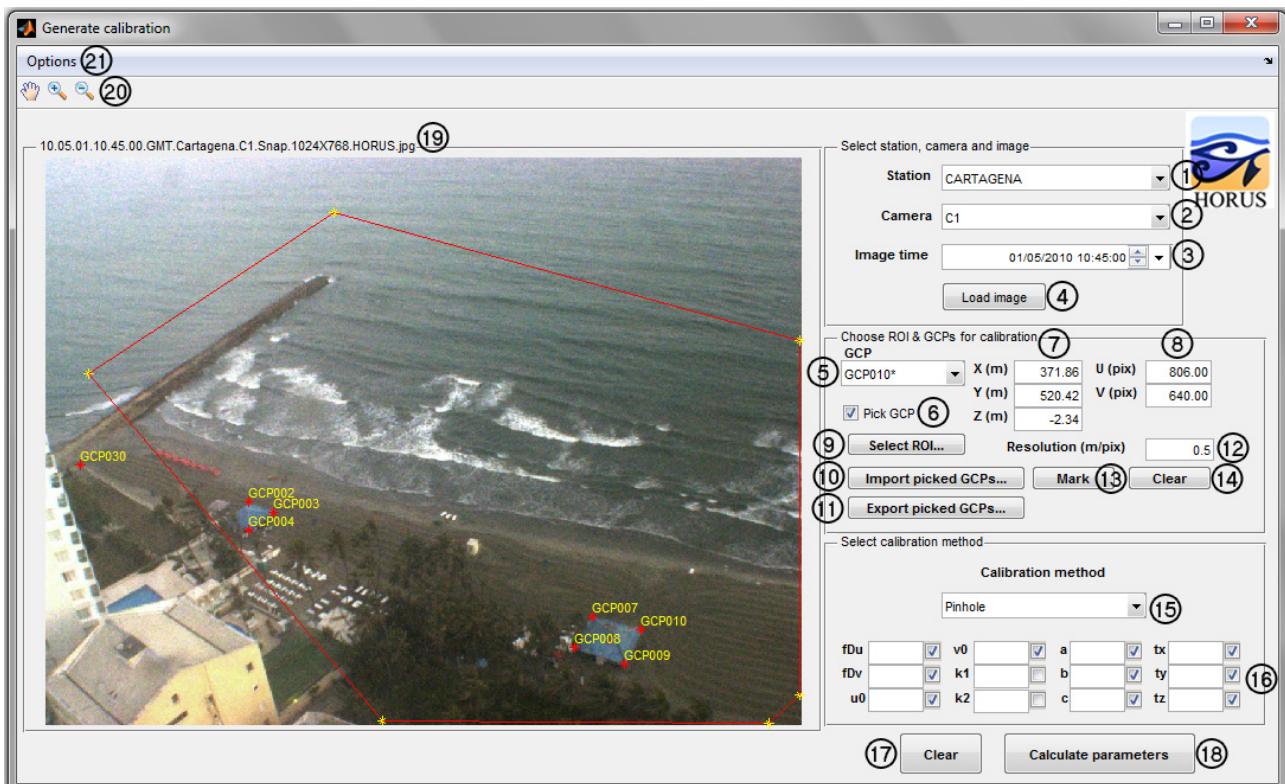


Figura 5.29: Interfaz de Calibración

El comando para ejecutar esta interfaz es: `gui_gencalibration`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable para seleccionar una estación.
2. Lista desplegable para seleccionar una cámara dada una estación. Esta lista se carga cada vez que se selecciona una estación.
3. Calendario donde se puede seleccionar un tiempo determinado para generar la calibración. Se carga la imagen más próxima en un rango de un día antes y después del tiempo seleccionado. La calibración que se quiere generar sirve para rectificar todas las imágenes que tengan un tiempo mayor a este tiempo escogido, hasta que se genere una nueva calibración, que se utilizaría para rectificar las imágenes subsiguientes.

4. Botón para cargar la imagen oblicua correspondiente a la estación, cámara y tiempo seleccionados, si ésta se encuentra presente en la base de datos y en la ruta especificada.
5. Lista desplegable donde se puede seleccionar un GCP específico en la estación. Si el GCP seleccionado en la lista es escogido para generar la calibración, junto a su nombre aparecerá un símbolo "\*" y se mostrará en la imagen como se muestra en la figura.
6. Escoger el GCP seleccionado para generar la calibración.
7. Coordenadas  $(x, y, z)$  del GCP seleccionado. Esta coordenada se carga desde la base de datos y cada GCP tiene asociada una.
8. Coordenadas  $(u, v)$  del GCP actual, si el usuario ya marcó estas coordenadas en la imagen.
9. Botón para seleccionar una ROI en la imagen. Este botón despliega la interfaz para seleccionar un ROI.
10. Botón para seleccionar un archivo donde está la información de los GCPs escogidos si existe de antemano. El archivo debe tener formato de EXCEL o MAT, y debe ser una matriz con tres columnas: el nombre del GCP (e.g. GCP001), coordenada en U y coordenada en V. Un ejemplo de archivo en formato EXCEL para importar los GCPs marcados se muestra en la figura 5.30.

	A	B	C
1	GCP030	48	416
2	GCP002	276	467
3	GCP003	309	481
4	GCP004	276	504
5	GCP007	741	621
6	GCP008	717	663
7	GCP009	785	686
8	GCP010	806	640

Figura 5.30: Ejemplo de archivo para importar GCPs marcados

11. Botón para exportar los GCPs marcados en la calibración actual a un archivo en formato de EXCEL o MAT. Este botón solamente se activa cuando haya puntos marcados en la imagen actual.
12. Resolución de la imagen rectificada en unidades de  $m/pix$ . Este parámetro es el número de metros en un píxel, se recomienda un valor de 0,3 en adelante.
13. Botón que activa la selección de las coordenadas  $(u, v)$  para el GCP seleccionado. Cuando esta opción se activa en el área donde se muestra la imagen el puntero cambia a forma de cruz y el usuario puede seleccionar un punto. Cuando el puntero se mueve, las coordenadas se actualizan en los campos de texto. Cuando el usuario escoge un punto al dar clic, al GCP se le asigna la coordenada  $(u, v)$  escogida.
14. Botón para limpiar la coordenada  $(u, v)$  marcada.
15. Lista desplegable donde se puede seleccionar el método de calibración (e.g. *DLT*, *RANSAC-DLT*, *Pinhole*).

16. Si el método escogido es *Pinhole*, se muestra la lista de estimados iniciales que corresponden a la última calibración almacenada en la base de datos. En caso contrario, estas opciones aparecen deshabilitadas. El usuario puede seleccionar cualquier subconjunto de estos parámetros como estimados iniciales para el método *Pinhole*. Se recomienda usar como estimados iniciales los parámetros calculados con el método *DLT*. Si ya se ha calculado una calibración anteriormente, automáticamente sus parámetros se convierten en los estimados iniciales para la siguiente calibración.
17. Botón para limpiar los campos mostrados en el numeral anterior.
18. Botón que comienza el proceso de calibración. Dependiendo del método de calibración escogido se generan distintos parámetros. Es recomendable utilizar el método *Pinhole* ya que es el más robusto y considera la distorsión generada en la imagen por los lentes de las cámaras. Al terminar el proceso de calibración, se muestran en la imagen las aproximaciones de los GCPs con el método y su distancia a los GCPs reales. En la línea de comandos se muestran la matriz de transformación y los errores cuadráticos medios en píxeles y en metros.
19. Área donde se pintan la imagen seleccionada, los GCPs escogidos y el ROI escogido. Por defecto, el ROI es el más cercano en tiempo antes del tiempo de la imagen, cargado desde la base de datos.
20. Controles para acercar, alejar o mover la imagen.
21. El usuario tiene la opción de guardar en la base de datos la calibración generada en el menú *Options ->Save calibration*. También puede ver en cualquier momento los estimados para los GCPs marcados en la imagen mediante el menú *Options ->Show calibration*. También puede ver en cualquier momento la imagen rectificada de prueba en el menú *Options ->Show rectified image*.

En la figura 5.31 se muestra el diagrama de flujo del proceso de generación de calibración donde para cada proceso se nombra la función que utiliza.

## 5.7. Create new ROI

Con esta interfaz se puede insertar, actualizar o eliminar la información en la base de datos de un ROI asociado a una calibración. Al abrir la interfaz, se muestran los ROIs disponibles en la base de datos, los cuales pueden ser actualizados. También es posible agregar uno nuevo.

En esta interfaz cada vez que se hace alguna consulta, se inserta, actualiza o elimina información de la base de datos, se mostrará una ventana indicando esto y se cerrará automáticamente cuando haya finalizado.

En la figura 5.32 se muestra la opción de insertar un ROI en la base de datos y en la figura 5.33 la opción de actualizar o eliminar.

El comando para ejecutar esta interfaz es: `gui_roi_tool`.

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. Lista para seleccionar la estación en la cual se encuentra el ROI a actualizar o en donde se desea insertar.
2. Lista para seleccionar la cámara en la cual se encuentra el ROI a actualizar o en donde se desea insertar.

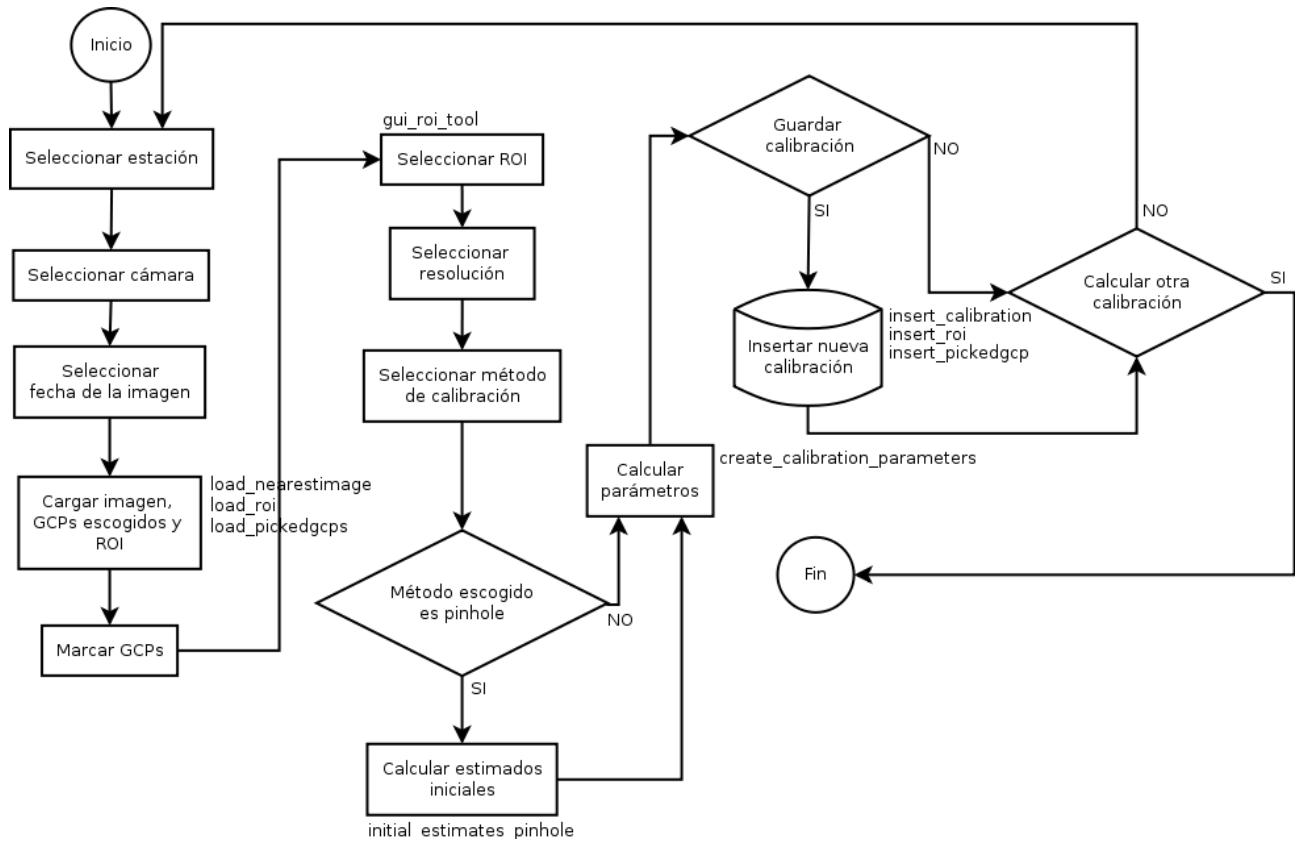


Figura 5.31: Diagrama de flujo del proceso de generación de calibración

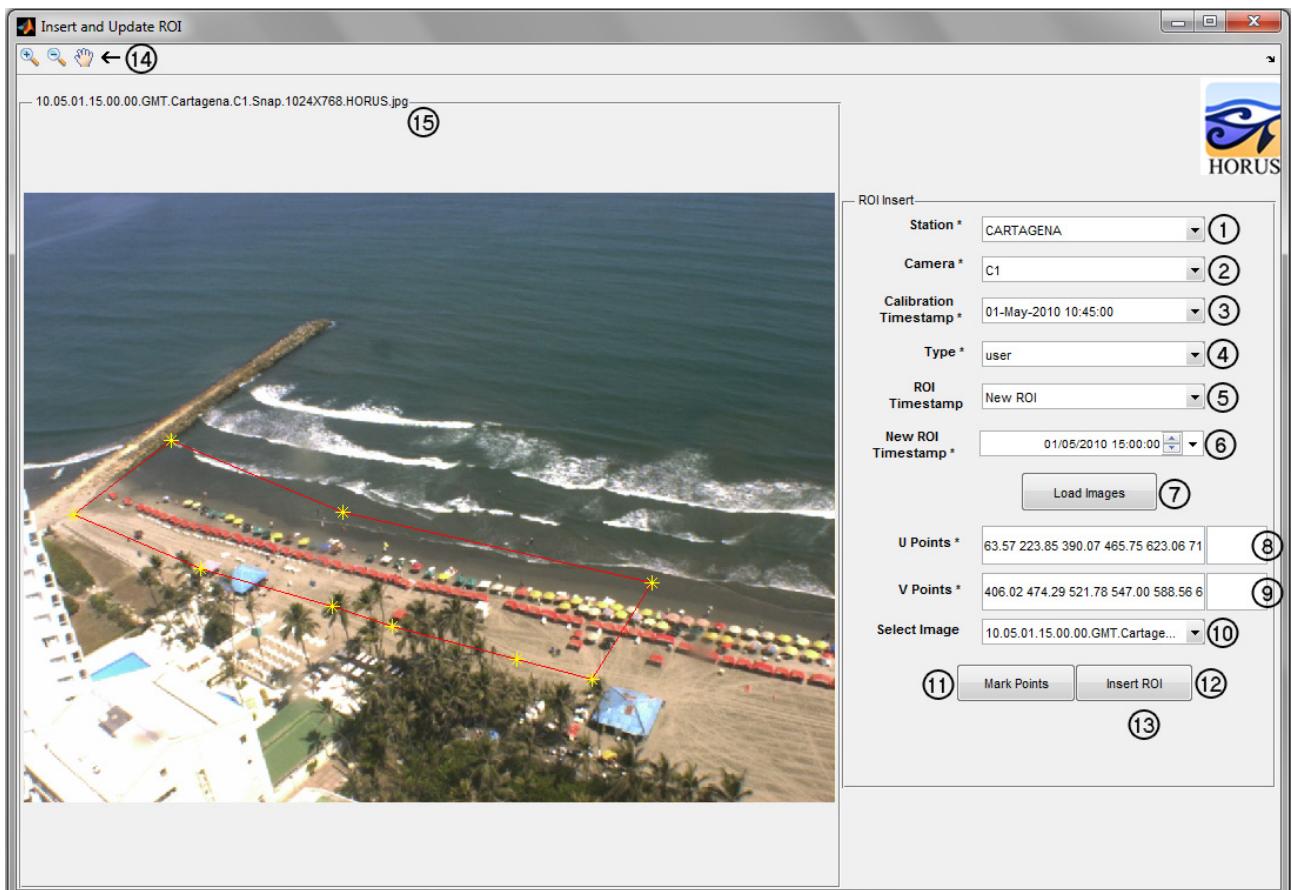


Figura 5.32: Interfaz para insertar un nuevo ROI

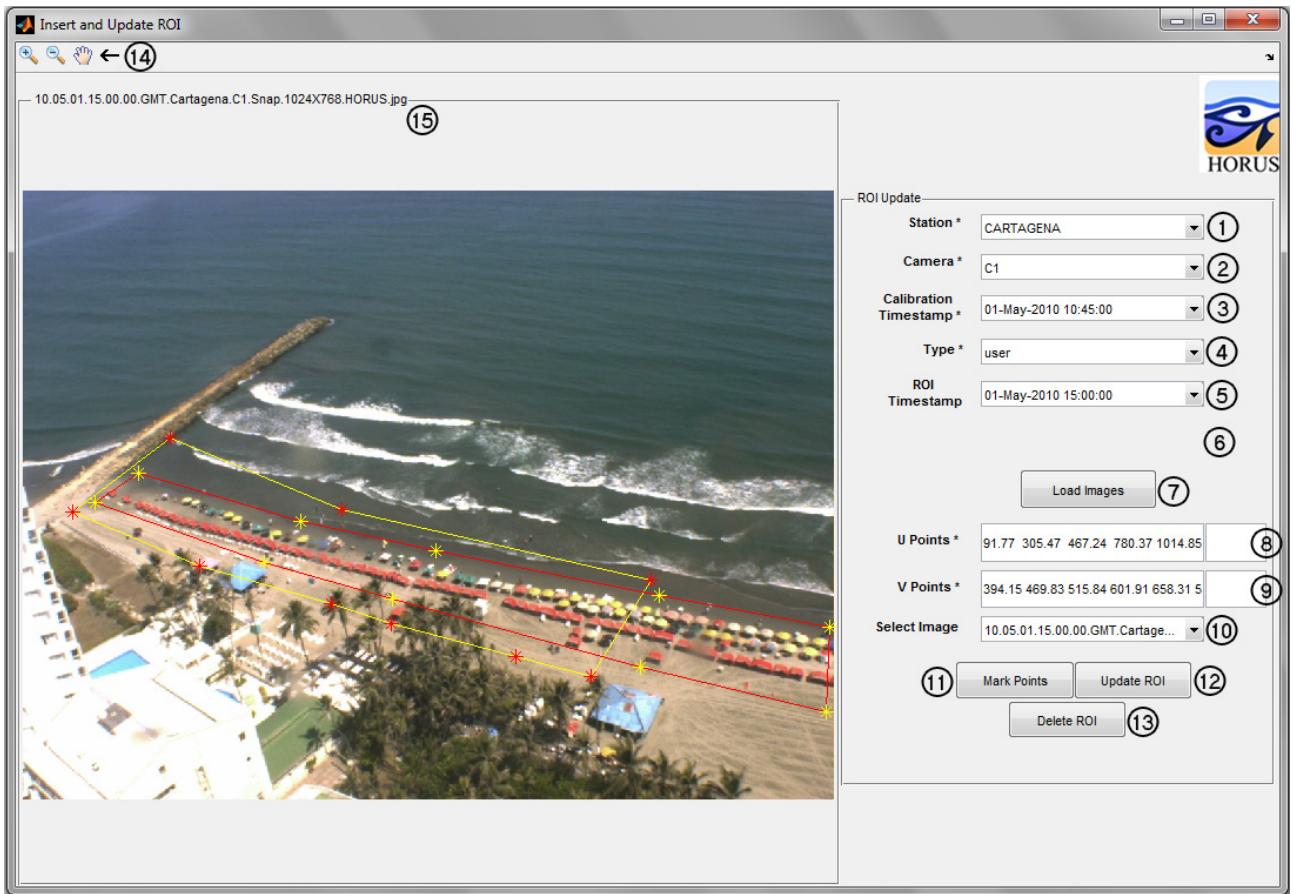


Figura 5.33: Interfaz para actualizar un ROI

3. Lista para seleccionar un tiempo de una calibración en el cual se desea actualizar o insertar el ROI.
4. Lista para seleccionar el tipo del ROI el cual puede ser *rect* (para la rectificación), *stack* (para *timestacks*) o *user* (para densidad de usuarios).
5. Tiempo del ROI que se desea actualizar. También se puede seleccionar insertar un nuevo ROI. Este tiempo es una fecha que sirve para definir a partir de qué momento es válido el ROI. Cuando se debe usar un ROI en algún algoritmo se toma el último que existe en la base de datos con un tiempo antes de la fecha de la imagen.
6. Si se va a insertar un nuevo ROI, esta opción se activará y se debe ingresar el tiempo del ROI.
7. Botón para cargar las imágenes candidatas para marcar los ROIs. Este botón debe ser presionado para poder visualizar una lista de imágenes en (10) para posteriormente ser mostrada una imagen en (15), si esto no se hace no se podrá mostrar la imagen. Las imágenes que se listan son las veinte primeras imágenes que existan a partir de la fecha que se muestra acá y hasta doce horas después de ésta.
8. Posiciones en *u* (valores reales), si se escogió la opción de actualizar y hay información en la base de datos, por el contrario, si se escogió la opción de insertar se mostrará vacío para que se ingrese la información. El campo que se muestra al lado derecho es para visualizar la posición de los puntos cuando se está haciendo la marcación con (11). Deben ser mínimo tres puntos para poder así cerrar el polígono.

9. Posiciones en  $v$  (valores reales), si se escogió la opción de actualizar y hay información en la base de datos, por el contrario, si se escogió insertar se mostrará vacío para que se ingrese la información. El campo que se muestra al lado derecho es para visualizar la posición de los puntos cuando se está haciendo la marcación con (11). Deben ser mínimo tres puntos para poder así cerrar el polígono.
10. Lista para seleccionar la imagen que se quiere mostrar en (15) y sobre la cual se hará la marcación.
11. Botón para comenzar las marcaciones de los puntos  $(u, v)$  sobre la imagen y que son mostrados en (8) y (9). La marcación de los vértices se logra haciendo clic izquierdo sobre la imagen y marcando el último punto con clic derecho para cerrar el polígono. Si se desean agregar más vértices después de cerrar el polígono se puede presionar de nuevo (11) y éste borrará la línea de cierre y se podrán agregar los nuevos puntos. Para eliminar un vértice se puede presionar el botón <- que se encuentra en (14).
12. Botón de insertar/actualizar un ROI que debe ser presionado para generar la acción, éste se habilita cuando se ha ingresado la información correctamente y se comprueba que no hayan campos vacíos en la parte obligatoria y sí sean numéricos los campos (8) y (9). Al terminar, se mostrará una ventana indicando el éxito o el fracaso de la operación.
13. Este botón aparece cuando se ha ingresado la información como si se fuese a actualizar un ROI y sirve para eliminarlo de la base de datos. Si se desea insertar un nuevo ROI, este botón no será visible. Al terminar, se mostrará una ventana indicando el éxito o el fracaso de la operación.
14. Opciones para hacer *zoom*, desplazarse sobre la imagen y borrar el último vértice marcado sobre la imagen (<-). Se puede eliminar cualquier cantidad de puntos.
15. Área donde se muestra una imagen en la que se marcan los puntos U y V del ROI. Cuando se activa la opción de insertar o actualizar un ROI, las líneas se verán rojas y los puntos amarillos. Cuando se está visualizando la anterior marcación (la que se va a actualizar) las líneas se ven amarillas y los puntos rojos.

Se puede hacer la marcación de dos formas: agregando los puntos en los campos de texto (8) y (9) o marcándolos en la imagen con (11). Se debe tener en cuenta que si se está actualizando, al presionar (11) por primera vez, se deberá empezar a marcar el polígono desde el principio. Si la idea es agregar sólo un punto, es recomendable hacerlo con (8) y (9), o marcar sobre los puntos anteriores. Estas dos formas no son excluyentes y se pueden combinar a gusto del usuario.

En la figura 5.34 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

## 5.8. Manually rectify images

La interfaz para la rectificación manual de lotes de imágenes tiene como fin proveer al usuario una manera de rectificar un conjunto de imágenes dados una estación, una cámara, un tipo de imagen y un rango de tiempo. El usuario puede definir un ROI en la imagen para la rectificación. Se presupone que ya hay calibraciones almacenadas en la base de datos, de lo contrario el usuario debe proceder a generar una con la interfaz correspondiente. Por defecto, se escoge la última calibración almacenada en la base de datos cuyo tiempo de generación sea el más cercano antes del tiempo de cada imagen, sin embargo, es posible escoger cualquier otra

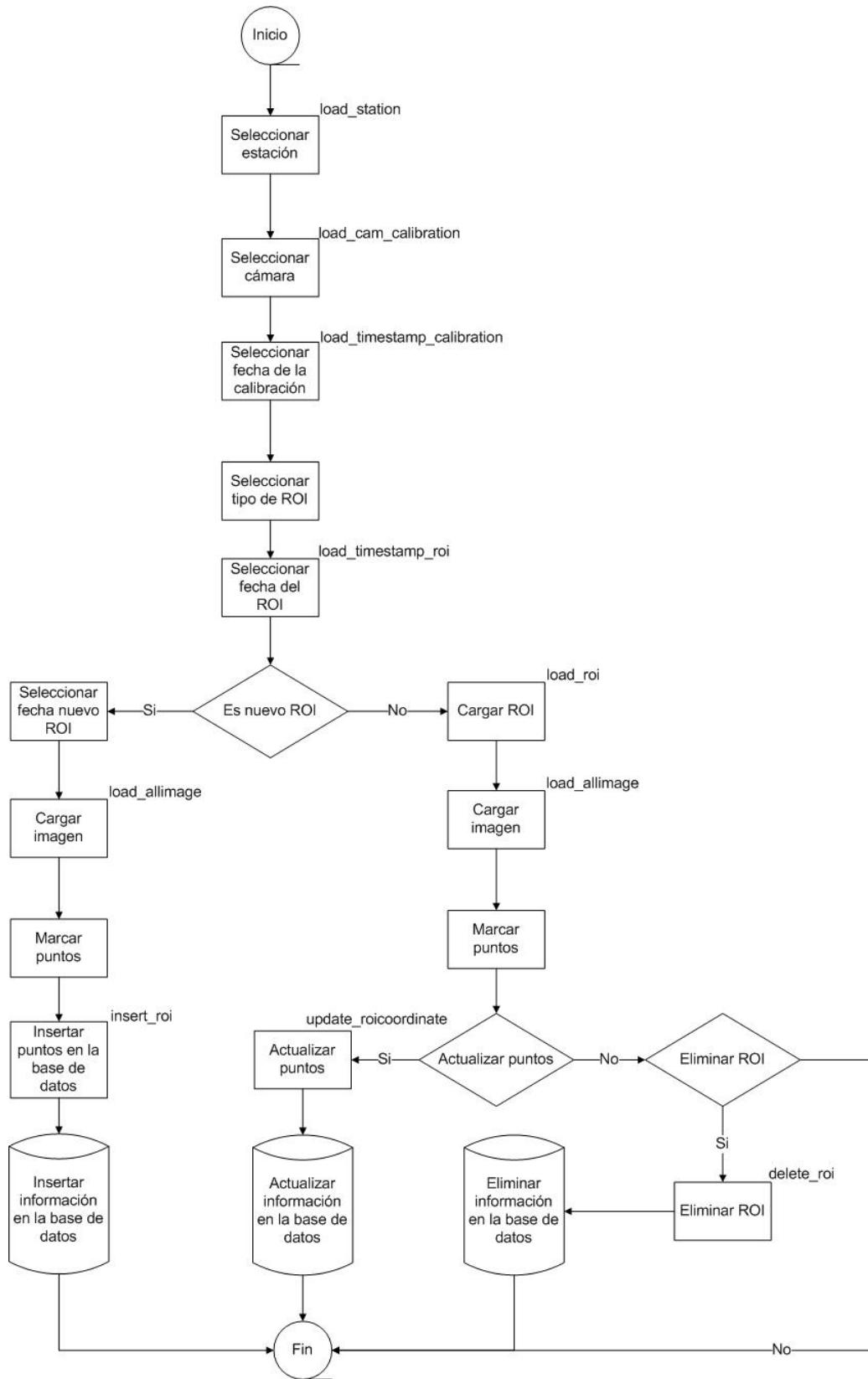


Figura 5.34: Diagrama de flujo interfaz ROI Tool

calibración para rectificar el conjunto completo de imágenes. En la figura 5.35 se muestra la interfaz.

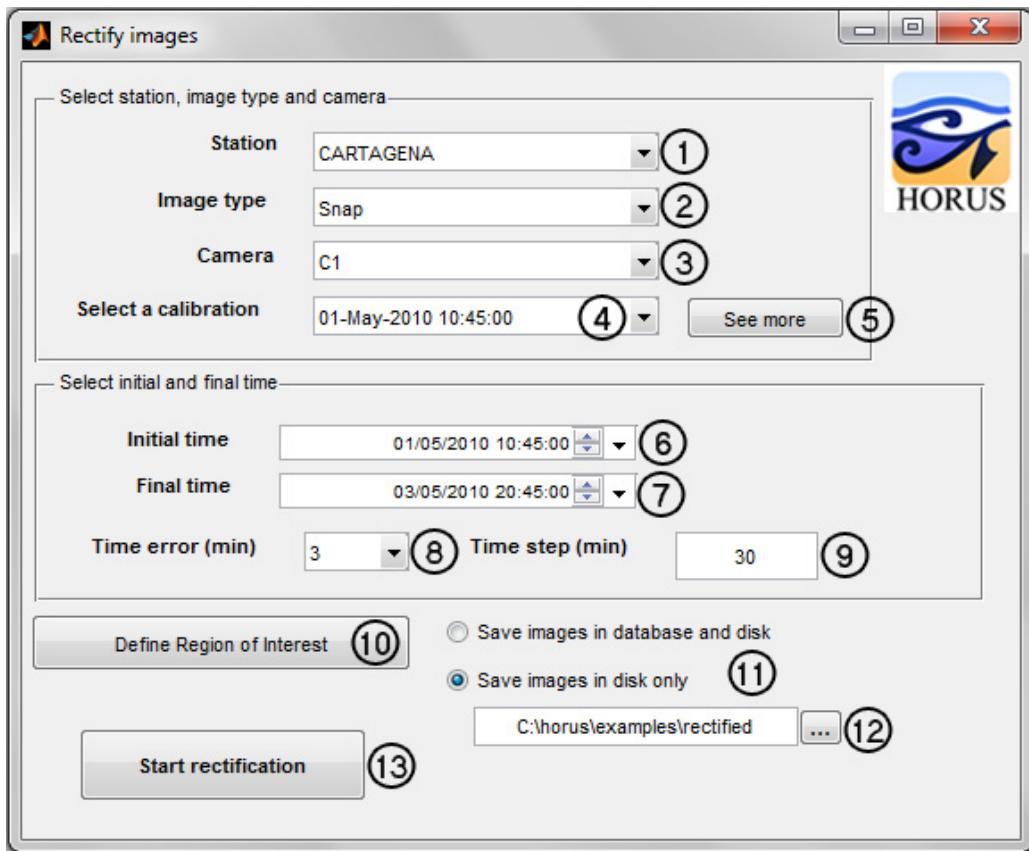


Figura 5.35: Interfaz de Rectificación

El comando para ejecutar esta interfaz es: `gui_rectify_images`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable donde se puede escoger una de las estaciones presentes en la base de datos.
2. Lista desplegable donde se puede escoger un tipo de imagen que esté disponible en la base de datos. (e.g. *Snap*, *Timex*, *Var*).
3. Lista desplegable donde se puede escoger una cámara de una estación. Las opciones sólo se habilitan si hay una estación seleccionada.
4. Lista desplegable donde se puede escoger la fecha de cualquier calibración almacenada en la base de datos para la estación y la cámara escogidas.
5. Botón para ver información detallada de la calibración escogida.
6. Calendario donde se puede seleccionar el tiempo de búsqueda inicial de imágenes para la rectificación. No se permite al usuario escoger un tiempo inicial mayor que el tiempo final actualmente seleccionado.
7. Calendario donde se puede seleccionar el tiempo de búsqueda final de imágenes para la rectificación. No se permite al usuario escoger un tiempo final menor que el tiempo inicial actualmente seleccionado.

8. Lista desplegable donde se puede seleccionar el error en el tiempo para buscar las imágenes, en minutos. Este error puede estar entre cero y diez minutos.
9. Área de texto donde el usuario puede ingresar el tamaño de paso para la búsqueda de imágenes (en minutos). La búsqueda de imágenes se realiza por el tiempo de la imagen. Sea  $t_i$  el tiempo inicial de búsqueda de imágenes,  $t_f$  el tiempo final de búsqueda de imágenes y  $\Delta t$  el tamaño de paso. Las imágenes que se van a considerar para la rectificación son aquellas que pertenecen a la estación, la cámara y son del tipo escogidos por el usuario, y además de eso, son las que tienen un tiempo  $t$ , y este tiempo pertenece al conjunto  $\{t : t \geq t_i \wedge t \leq t_f \wedge t = t_i + k \cdot \Delta t\}$ , donde  $k \in \{0, 1, 2, \dots\}$ .
10. Botón que despliega una interfaz para seleccionar un ROI en la cámara para la rectificación de las imágenes.
11. El usuario tiene la opción de guardar las imágenes generadas por el proceso de rectificación en la base de datos y en la ruta en el disco para almacenar estas imágenes por defecto (esta ruta es escogida por el usuario cuando configura por primera vez una estación, y se almacena en `path_info.xml`), o puede seleccionar una ruta diferente en el disco, en caso de que no quiera guardar estas imágenes en la base de datos.
12. Botón para seleccionar la ruta del nuevo directorio donde se guardarán las imágenes rectificadas.
13. Luego de que todos los parámetros de la rectificación se han seleccionado, este botón comienza el proceso de rectificación. Se buscan las imágenes en la base de datos, se carga la calibración más cercana antes del tiempo de búsqueda inicial de imágenes, y se aplica la rectificación a cada una de las imágenes encontradas de manera secuencial. Cada imagen se almacena en el disco dependiendo de la ruta que el usuario haya escogido. Si se escoge guardar en la base de datos, las imágenes generadas se almacenan en la ruta por defecto, de lo contrario se almacenan en la ruta que el usuario haya escogido.

En la figura 5.36 se muestra el diagrama de flujo para el proceso de rectificación en lote, donde para cada proceso se nombra la función que utiliza.

## 5.9. Configure fusion

La interfaz de calibración de fusión provee al usuario una manera de generar los parámetros del modelo que sirve para fusionar un grupo de imágenes. El usuario selecciona una estación y un tiempo. Hay dos tipos de calibraciones, una para fusionar imágenes oblicuas y otra para fusionar imágenes rectificadas. Las imágenes para un mismo tiempo y diferentes cámaras pueden tener una diferencia debido a la no simultaneidad en la captura, el usuario puede definir un margen de error para esta diferencia a la hora de hacer la búsqueda de imágenes en la base de datos. Se selecciona el conjunto de cámaras que participan en la fusión y el orden entre ellas. Por último, para cada par de imágenes correspondientes a cámaras consecutivas en el orden, el usuario puede marcar los puntos en común que se mapean a una coordenada  $(x, y, z)$  conocida. El proceso se repite para todos los pares de cámaras en el orden establecido. Al final, se calculan los parámetros del modelo que sirven para fusionar las imágenes. En la figura 5.37 se muestra la interfaz.

El comando para ejecutar esta interfaz es: `gui_genfusion`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable para seleccionar una estación.

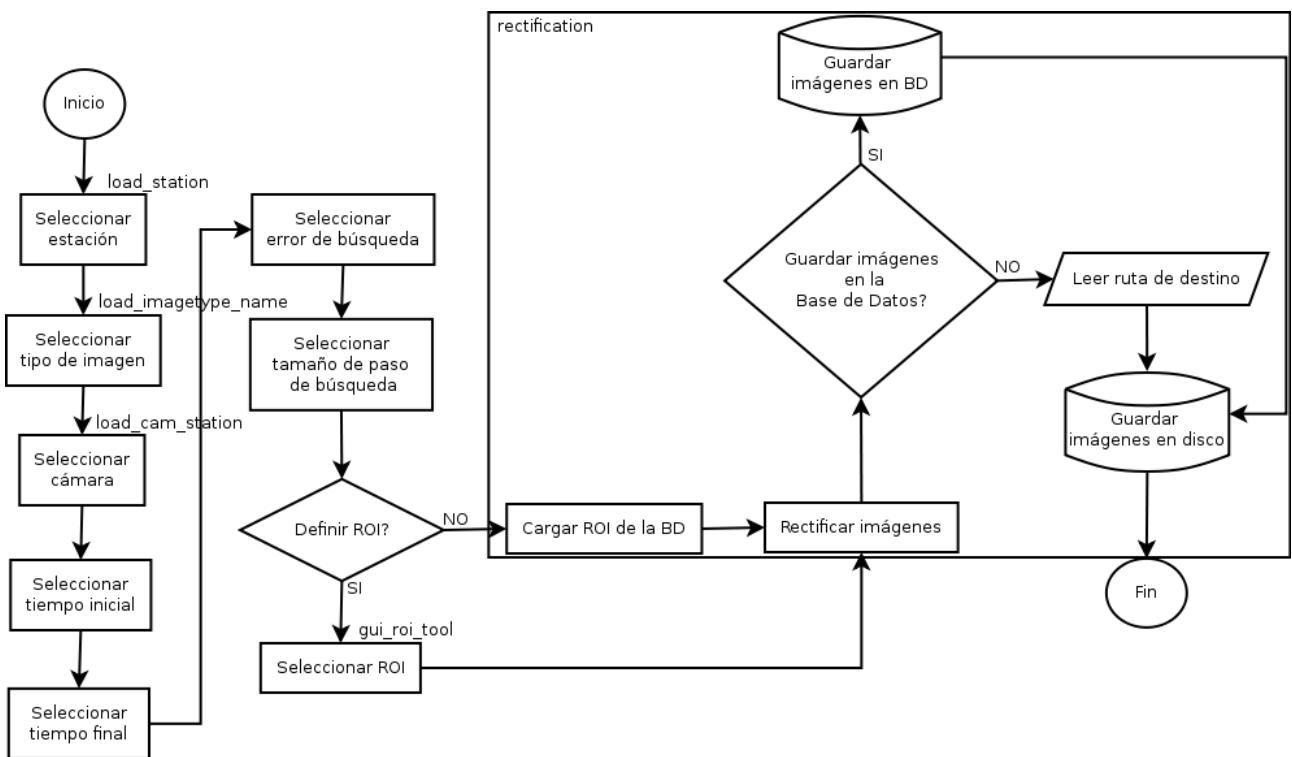


Figura 5.36: Diagrama de flujo del proceso de rectificación en lote

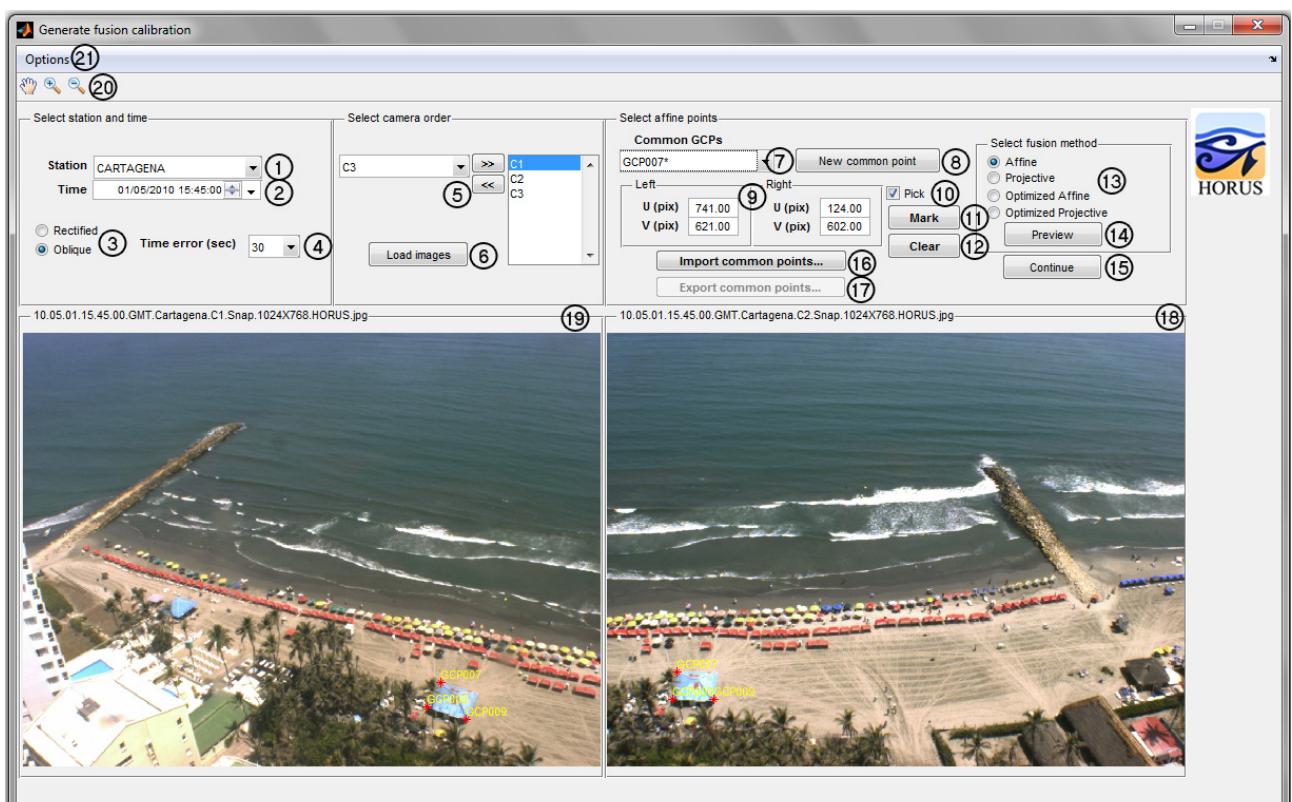


Figura 5.37: Interfaz de Calibración de fusión

2. Calendario donde se puede seleccionar un tiempo determinado para generar la calibración. Es necesario escoger un tiempo donde haya imágenes para todas las cámaras que se vayan a elegir.
3. Opción para escoger una fusión de imágenes oblicuas o rectificadas.
4. Margen de error en segundos para la diferencia en el tiempo de las imágenes de diferentes cámaras.
5. Selección de las cámaras y el orden entre ellas. En la lista desplegable se listan todas las cámaras de la estación escogida, y mediante el botón “>>” se cargan para la calibración. El botón “<<” borra de la lista de cámaras la cámara seleccionada.
6. Botón para cargar las imágenes de las cámaras escogidas. Estas imágenes se cargan por pares, por ejemplo, si el usuario seleccionó las cámaras  $C_1$ ,  $C_2$  y  $C_3$ , primero se cargan las imágenes de las cámaras  $C_1$  y  $C_2$ , se generan los parámetros para este par, y luego se cargan las imágenes de las cámaras  $C_2$  y  $C_3$ .
7. Lista desplegable donde aparecen todos los puntos de control comunes para el par de imágenes actuales. Si un punto se escoge para la calibración, junto a su nombre aparece un “\*” y se dibuja en las imágenes correspondientes.
8. Botón para agregar un punto de control común. Cuando el usuario presiona este botón debe marcar el punto en la imagen de la izquierda, y luego marcar el mismo punto en la imagen de la derecha. El nombre del nuevo punto es de la forma “NEWXXX” donde “XXX” es un consecutivo de acuerdo al número de puntos nuevos que han sido generados.
9. Posiciones  $(u, v)$  (ubicación dentro de la imagen) para el punto escogido en la imagen izquierda (*Left*) y en la imagen derecha (*Right*).
10. Opción para escoger o no un punto de control común en la calibración.
11. Botón para marcar los puntos  $(u, v)$  del punto de control seleccionado. Primero se marca un punto de la imagen izquierda, y luego se marca ese mismo punto en la imagen derecha.
12. Botón para limpiar las coordenadas del punto de control seleccionado.
13. Tipos de métodos para calcular los parámetros de fusión cuando la fusión es para imágenes oblicuas. Se puede seleccionar uno de los siguientes cuatro tipos de fusión: *Affine*, *Projective*, *Optimized Affine* y *Optimized Projective*.
14. Botón para previsualizar la imagen fusionada con las dos imágenes seleccionadas, utilizando el método seleccionado en el paso anterior. Esto tiene la finalidad de que el usuario pueda seleccionar de manera visual el método más adecuado para la fusión.
15. Botón para calcular los parámetros de fusión del par de cámaras actuales. Si aún hay cámaras por procesar, el botón contiene el texto “Continue”, calcula los parámetros con el método seleccionado y carga el siguiente par de imágenes. Si ya no hay más cámaras por procesar, el botón contiene el texto “Calculate”, calcula los parámetros y termina. Al final, se muestra una imagen fusionada con los parámetros calculados que tiene como finalidad validar el correcto funcionamiento.
16. Botón para seleccionar un archivo donde está la información de los puntos comunes marcados, si existe de antemano. El archivo debe tener formato EXCEL, MAT o TXT, y debe ser una matriz con cuatro columnas: ID de la cámara en la que está marcado cada punto

(e.g. C1), el nombre del punto (e.g. PNT00001), la coordenada en U y la coordenada en V. Un ejemplo de archivo en formato EXCEL para importar los puntos comunes marcados se muestra en la figura 5.38.

	A	B	C	D
1	C1	GCP007	741	621
2	C2	GCP007	124	602
3	C1	GCP008	717	663
4	C2	GCP008	116	651
5	C1	GCP009	785	686
6	C2	GCP009	189	651
7	C1	NEW001	807.347603	642.761886
8	C2	NEW001	196.295194	604.034765
9	C2	GCP014	957	625
10	C3	GCP014	371	713
11	C2	GCP015	924	585
12	C3	GCP015	330	687
13	C2	GCP016	909	503
14	C3	GCP016	295	613

Figura 5.38: Ejemplo de archivo para importar puntos comunes marcados

17. Botón para exportar los puntos comunes marcado para todas las cámaras, en el mismo formato del numeral anterior. Este botón sólo se activa cuando se han terminado de marcar todas las cámaras y se han calculado los parámetros.
18. Imagen correspondiente a la cámara derecha (El proceso de calibración se realiza de izquierda a derecha).
19. Imagen correspondiente a la cámara izquierda (El proceso de calibración se realiza de izquierda a derecha).
20. Controles para acercar, alejar o mover la imagen.
21. El usuario tiene la opción de guardar en la base de datos la calibración generada en el menú *Options ->Save fusion calibration*. También puede ver en cualquier momento la imagen fusionada de prueba *Options ->Show merged image*.

En la figura 5.39 se muestra el diagrama de flujo del proceso de generación de los parámetros fusión. El texto afuera de los recuadros es la función utilizada en ese proceso.

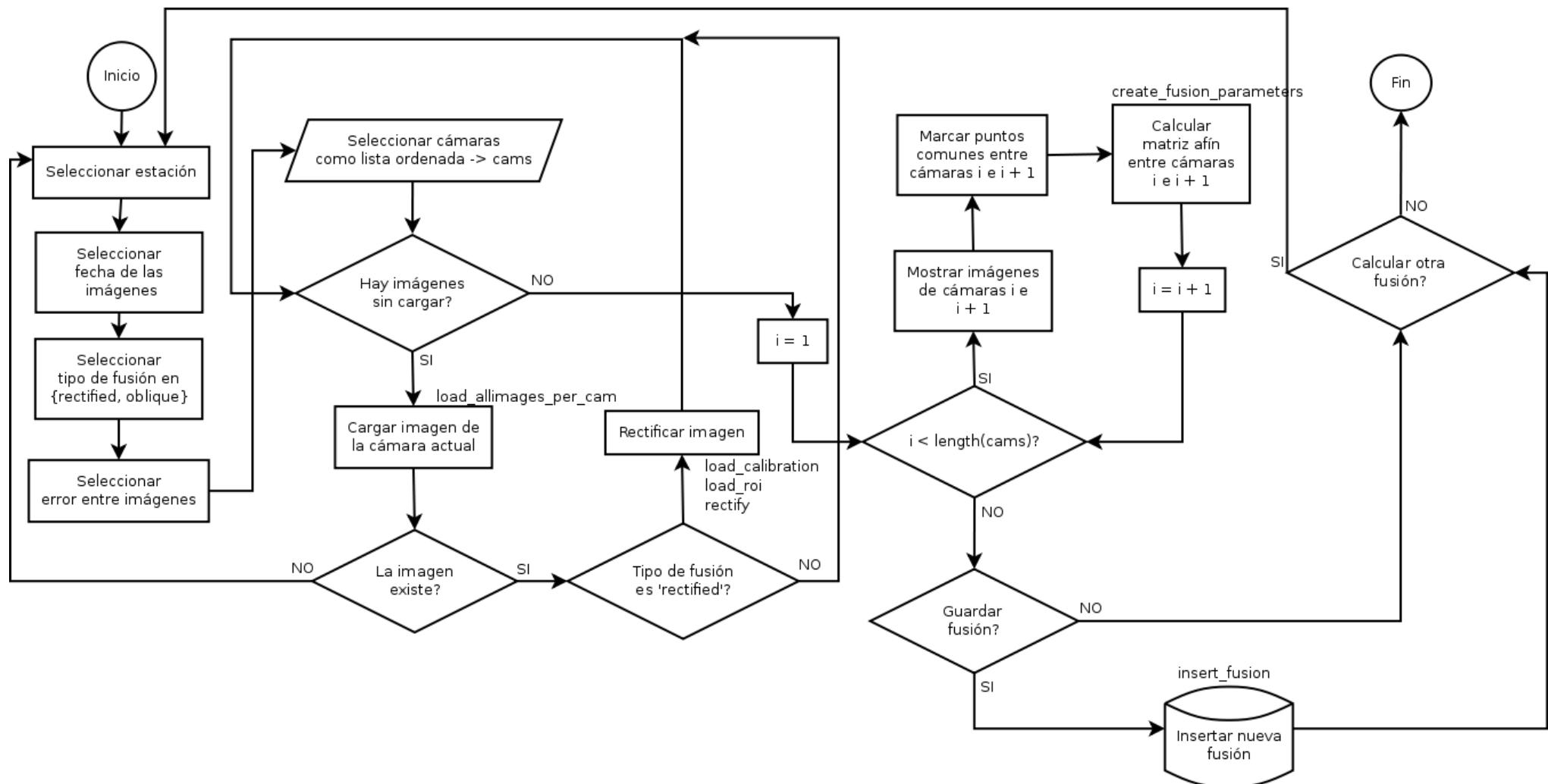


Figura 5.39: Diagrama de flujo del proceso de generación de parámetros de fusión

## 5.10. Manually merge images

La interfaz para la fusión de un lote de imágenes manualmente tiene como fin proveer al usuario una manera de configurar los criterios de búsqueda de un grupo de imágenes y fusionarlas dados una estación, un tipo de imagen y un rango de tiempo. El usuario tiene la opción de fusionar imágenes oblicuas, fusionar imágenes ya rectificadas, o de rectificar y luego fusionar las imágenes, para un conjunto de cámaras. Por defecto, se escoge la última calibración almacenada en la base de datos cuyo tiempo de generación sea el más cercano antes del tiempo de cada conjunto de imágenes, sin embargo, es posible escoger cualquier otra calibración para fusionar el conjunto completo de imágenes. En la figura 5.40 se muestra la interfaz.

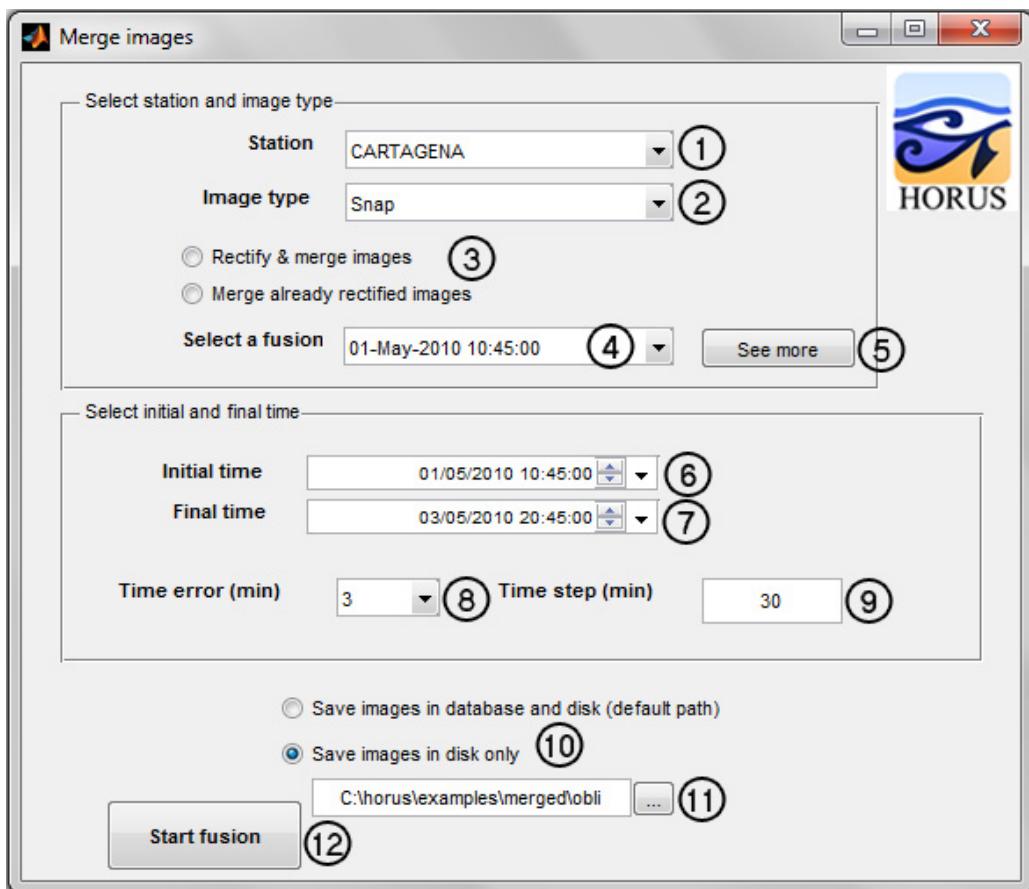


Figura 5.40: Interfaz de Fusión

El comando para ejecutar esta interfaz es: `gui_merging_images`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable donde se puede escoger una de las estaciones presentes en la base de datos.
2. Lista desplegable donde se puede escoger alguno de los tipos de imágenes disponibles en la base de datos (e.g.*Snap*, *Timex*, *Var*).
3. Se puede seleccionar una de las siguientes tres opciones: “Merge oblique images”, “Rectify and merge oblique images” y “Merge rectified images”.
4. Lista desplegable donde se puede escoger la fecha de una fusión almacenada en la base de datos para fusionar todo el conjunto de imágenes. Si se deja la opción por defecto, para cada imagen se busca la fusión más cercana antes del tiempo de la imagen.

5. Botón para ver más información sobre la fusión seleccionada.
6. Calendario donde se puede seleccionar el tiempo de búsqueda inicial de imágenes para la fusión. No se permite seleccionar un tiempo inicial mayor que el tiempo final seleccionado.
7. Calendario donde se puede seleccionar el tiempo de búsqueda final de imágenes para la fusión. No se permite seleccionar un tiempo final menor que el tiempo inicial seleccionado.
8. Lista desplegable donde se puede seleccionar el margen de error para la búsqueda de imágenes en un tiempo determinado para todas las cámaras que participan en la fusión. Puede tomar valores desde 0 min hasta 10 min. Esta opción es necesaria en ciertos casos donde las imágenes para un tiempo determinado no fueron capturadas de manera simultánea.
9. Área de texto donde el usuario puede ingresar el tamaño de paso para la búsqueda de imágenes (en minutos). La búsqueda de imágenes se realiza por el tiempo de la imagen.  
Sea  $t_i$  el tiempo inicial de búsqueda de imágenes,  $t_f$  el tiempo final de búsqueda de imágenes y  $\Delta t$  el tamaño de paso. Las imágenes que se van a considerar para la fusión son aquellas que pertenecen a la estación, la cámara y son del tipo escogidos por el usuario, y además de eso, son las que tienen un tiempo  $t$ , y este tiempo pertenece al conjunto  $\{t : t \geq t_i \wedge t \leq t_f \wedge t = t_i + k \cdot \Delta t\}$ , donde  $k \in \{0, 1, 2, \dots\}$ .
10. El usuario tiene la opción de guardar las imágenes generadas por el proceso de fusión en la base de datos y en la ruta en el disco para almacenar estas imágenes por defecto (esta ruta es escogida por el usuario cuando configura por primera vez una estación, y se almacena en `path_info.xml`), o puede seleccionar una ruta diferente en el disco, en caso de que no quiera guardar estas imágenes en la base de datos.
11. Botón para seleccionar la ruta del nuevo directorio donde se guardarán las imágenes fusionadas.
12. Cuando el usuario ya ha seleccionado todos los criterios de búsqueda de imágenes para fusionar, este botón comienza el proceso de fusión. La fusión se realiza utilizando los parámetros de fusión almacenados en la base de datos más cercanos antes del tiempo del conjunto de imágenes, así como la secuencia de cámaras que participan en la fusión.

En la figura 5.41 se muestra el diagrama de flujo del proceso de fusión en lote, donde para cada proceso se nombra la función que utiliza.

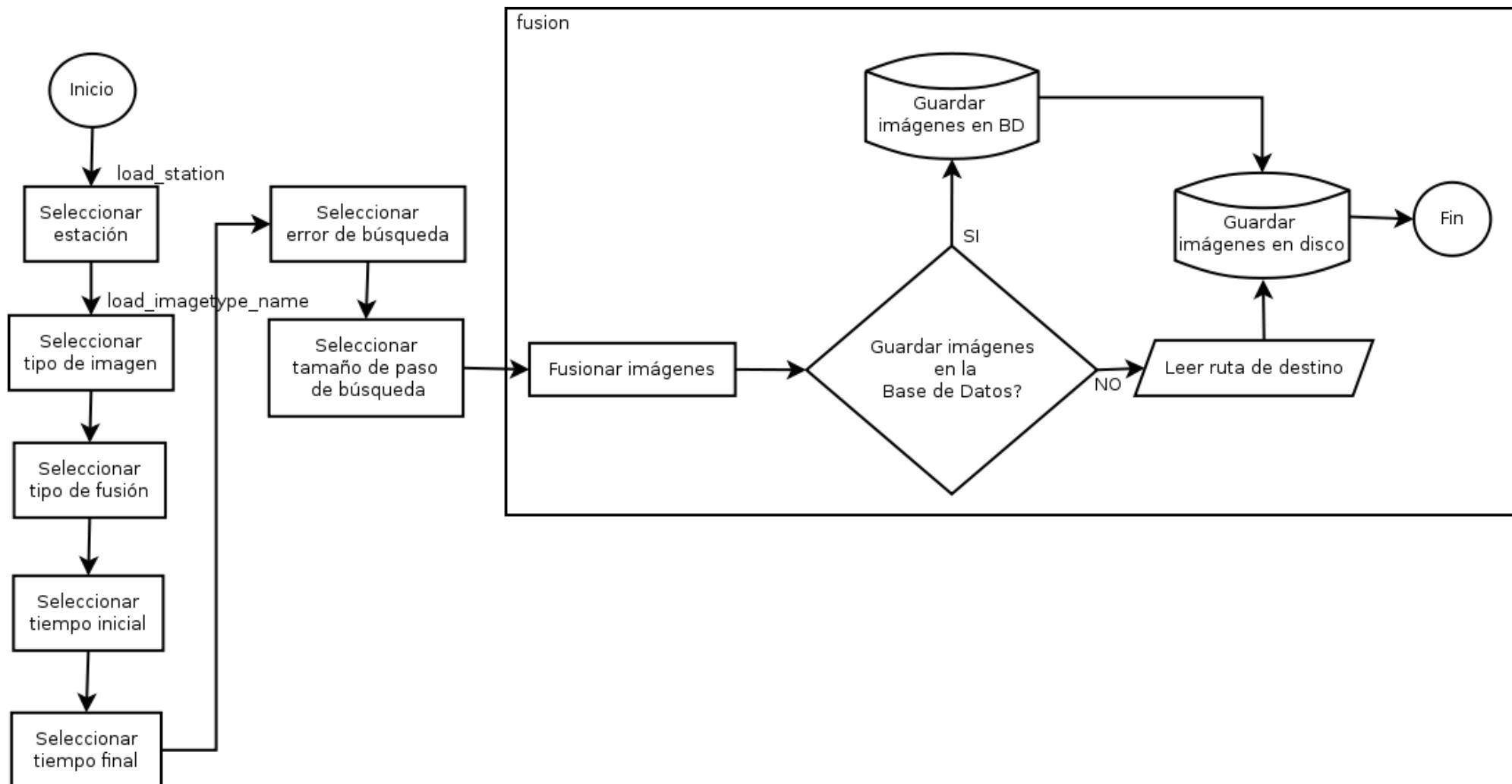


Figura 5.41: Diagrama de flujo del proceso de fusión en lote

## 5.11. Configure capture automatic

En esta sección, se muestra la interfaz para configurar la captura de imágenes oblicuas. Entre las cosas que se configuran se encuentran las horas del día en que se capturan imágenes y *stacks*, así como la información para el envío de los datos. La configuración se guarda en el archivo XML `capture_info.xml` que utiliza el programa automático de captura.

En la figura 5.42 se muestra la interfaz de configuración de captura.

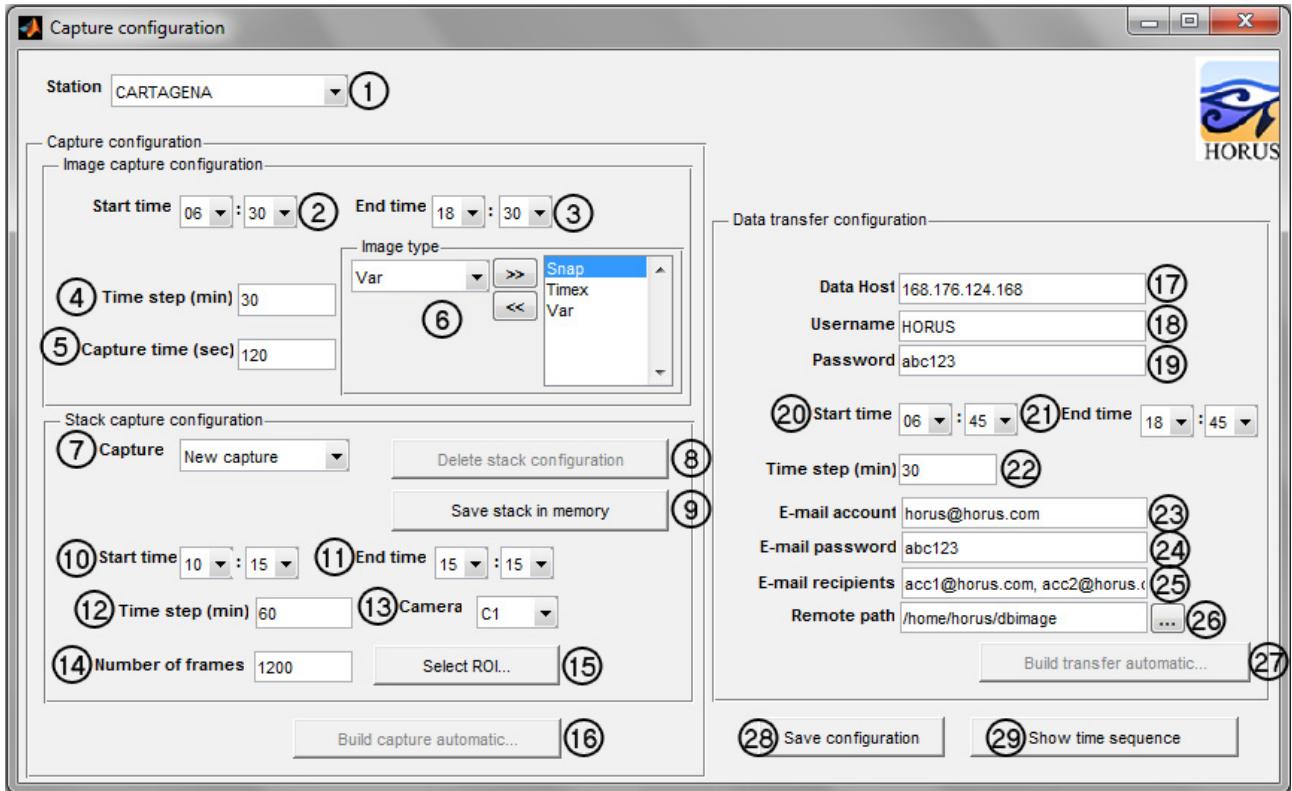


Figura 5.42: Interfaz de Configuración de capturas

El comando para ejecutar esta interfaz es: `gui_configure_capture`. Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable para seleccionar la estación.
2. Tiempo inicial de la captura de imágenes.
3. Tiempo final de la captura de imágenes.
4. Tiempo de espera entre capturas consecutivas, en minutos.
5. Tiempo de captura, en segundos.
6. Tipos de imágenes que se desean capturar.
7. Lista desplegable donde se puede seleccionar una configuración de captura de *stacks* para actualización o la opción de crear una nueva configuración.
8. Botón para eliminar la configuración de *stacks* seleccionada (sólo se habilita si hay una configuración seleccionada).

9. Botón para guardar en memoria la configuración de *stacks* actual. Esta configuración solamente se guarda en el archivo `capture_info.xml` al presionar el botón *Save configuration*.
10. Tiempo inicial de la captura de *stacks*.
11. Tiempo final de la captura de *stacks*.
12. Tamaño de paso entre captura y captura, en minutos.
13. Lista de las cámaras guardadas en la base de datos, se debe seleccionar una para asociarla a la captura.
14. Número de frames del vídeo. El *framerate* de la captura es el mismo que se eligió al configurar la cámara.
15. Botón para escoger una región de interés para la captura. La región de interés se define como el conjunto de coordenadas  $(u, v)$  de un polígono cerrado dentro de la imagen, en sentido horario o antihorario. Para seleccionar un ROI, se debe haber configurado la cámara seleccionada de antemano; al presionar este botón, la cámara captura una imagen y la muestra para seleccionar los puntos correspondientes a los vértices del polígono que conforma el ROI.
16. Botón para generar el automático de captura de imágenes y *stacks*.
17. La configuración del envío de datos contiene varios campos. Este campo es la dirección del servidor SSH al que se envían las imágenes.
18. Nombre de usuario del cliente SSH.
19. Contraseña del cliente SSH.
20. Tiempo inicial del envío.
21. Tiempo final del envío.
22. Tiempo de espera entre envíos consecutivos, en minutos.
23. Cuenta de e-mail desde donde se envían mensajes de error sobre el envío de datos.
24. Contraseña de la cuenta de e-mail.
25. Lista de e-mails separados por comas que reciben los mensajes de error.
26. Ruta remota en el servidor en donde se almacenarán los datos enviados.
27. Botón para generar el automático de transferencia.
28. Botón para almacenar la configuración en el archivo XML y en la base de datos (si todos los campos son válidos y no hay solapamiento entre tiempos de captura).
29. Botón que muestra gráficamente los tiempos de captura, transferencia y procesamiento previamente configurados.

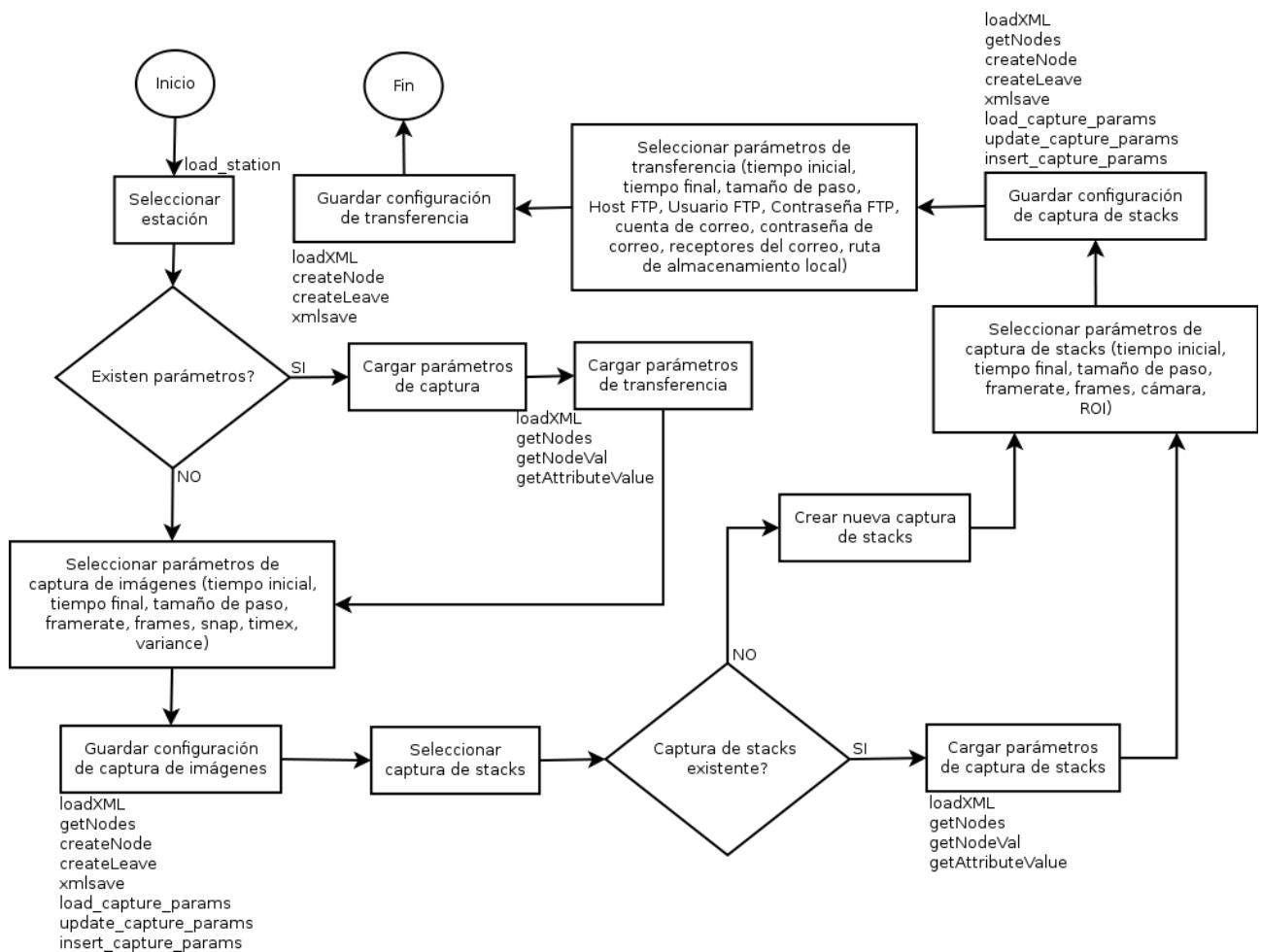


Figura 5.43: Diagrama de flujo del proceso de configuración de captura

En la figura 5.43 se muestra el diagrama de flujo para la configuración de capturas, donde para cada proceso se nombra la función que utiliza.

Toda esta información es utilizada por el algoritmo automático que lleva a cabo la captura. En la figura 5.44 se muestra el diagrama de flujo que describe el proceso de este automático. El automático se ejecuta como un servicio o demonio, es decir, se ejecuta infinitamente hasta que ocurre un evento que dispara una acción, en este caso, ese evento son las horas programadas para realizar la captura.

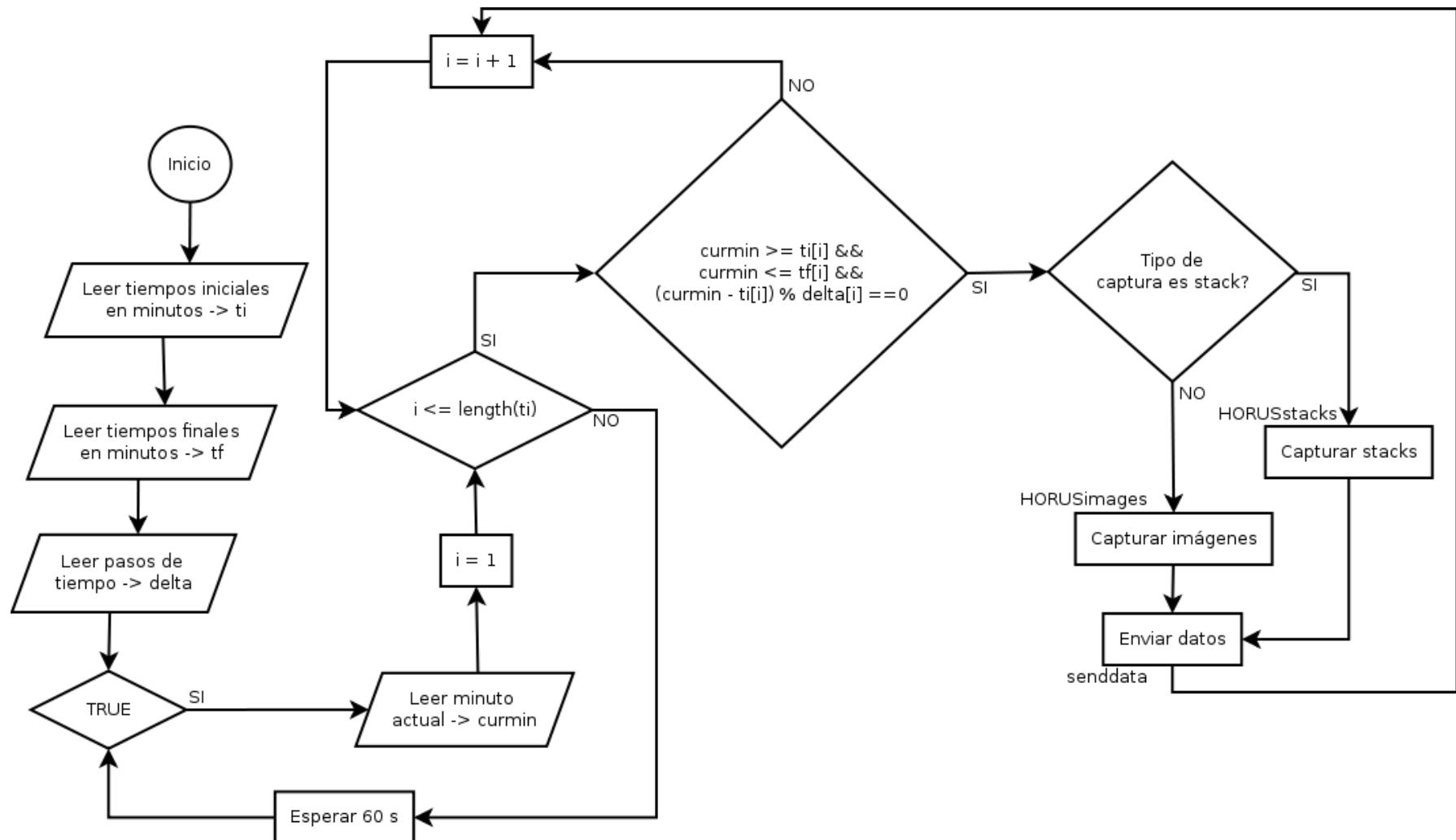


Figura 5.44: Diagrama de flujo del automático de captura

Para utilizar el automático de captura, es necesario compilar las funciones `auto_capture_images.m` y `auto_transfer.m` del directorio `src`, mediante los botones correspondientes en la interfaz. Sin embargo, si se quieren compilar manualmente, se ejecutan los siguientes comandos en la línea de comandos de MATLAB (suponiendo que el directorio de HORUS es `C:\horus`):

```
cd C:\horus
addpath('src')
mcc -m auto_capture_images -a ./data -I ./tmp
mcc -m auto_transfer -a ./data -I ./tmp
```

Si la compilación se realizó mediante la interfaz, se generan los archivos ejecutables `run_auto_capture_i` y `run_auto_transfer.bat`.

Si los archivos se compilaron manualmente, en el sistema operativo Windows, se generan dos ejecutables `auto_capture_images.exe` y `auto_transfer.exe`, los cuales reciben unos parámetros de entrada. El automático de captura recibe el nombre de la estación, y el automático de transferencia recibe el nombre de la estación y la dirección IP del servidor al que se enviarán los datos. Para ejecutarlos, se debe crear para cada uno, un archivo BAT que recibe los parámetros. Un ejemplo de archivo BAT para el automático de captura podría contener el siguiente código (`run_auto_capture_images.bat`):

```
auto_capture_images.exe CARTAGENA
```

Para el automático de transferencia (`run_auto_transfer.bat`):

```
auto_transfer.exe CARTAGENA 168.176.124.165
```

*Nota:* Es posible que al compilar el automático de transferencia aparezca un mensaje de advertencia. Esto es normal y se puede hacer caso omiso de él.

## 5.12. Configure processing automatic

En las secciones anteriores se presentaron las interfaces que permiten rectificar y fusionar imágenes manualmente. Sin embargo, en general es deseable que estos procesos se hagan automáticamente. En esta sección se describe la interfaz para configurar el automático de procesamiento para rectificar, fusionar, miniaturizar y enviar a la web las imágenes. La información de la configuración se guarda en el archivo XML `processing_info.xml`.

En la figura 5.45 se muestra la interfaz para configurar el automático de procesamiento.

El comando para ejecutar esta interfaz es: `gui_processing`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista desplegable para seleccionar la estación.
2. Tiempo inicial de la ejecución.
3. Tiempo final de la ejecución.
4. Tiempo de espera entre procesamientos consecutivos, en minutos.
5. Tipo de procesamiento a aplicar a las imágenes, las opciones son *Merge oblique images*, para fusionar imágenes oblicuas, *Rectify and merge oblique images* para rectificar y luego fusionar. *Rectify oblique images* para rectificar sin fusionar.
6. Tipos de imágenes que se van a procesar.

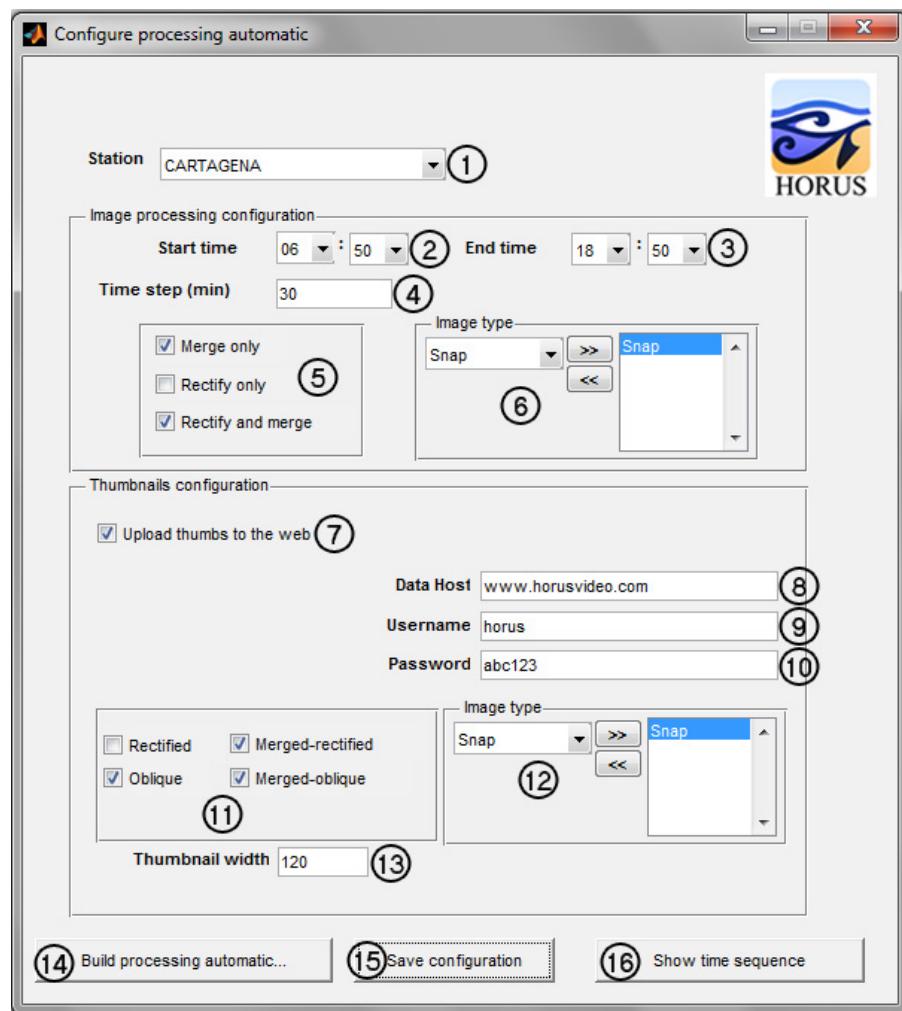


Figura 5.45: Interfaz para configurar el automático de procesamiento

7. Esta opción se habilita cuando se quiere enviar las miniaturas por Internet a un servidor web.
8. Nombre del servidor.
9. Nombre del usuario para acceder al servidor.
10. Contraseña para acceder al servidor.
11. Para ser visualizadas las imágenes procesadas en la web, es necesario generar miniaturas de estas imágenes. Aquí se seleccionan los tipos de imágenes procesadas que se van a miniaturizar.
12. Tipos de las imágenes que fueron procesadas.
13. Ancho deseado de la miniatura. Se calcula el tamaño para la miniatura de tal manera que el ancho sea el especificado en este campo, y la altura sea proporcional.
14. Botón para generar el automático de procesamiento.
15. Botón para almacenar la configuración del automático en el archivo XML y en la base de datos.
16. Botón para mostrar de manera gráfica los tiempos de captura y procesamiento previamente configurados.

En la figura 5.46 se muestra el diagrama de flujo del proceso de configuración de procesamiento de imágenes, donde para cada proceso se nombra la función que utiliza.

Esta información es utilizada por el automático de procesamiento en su ejecución. En la figura 5.47 se muestra el diagrama de flujo del algoritmo automático. El automático se ejecuta como un servicio o demonio, es decir, se ejecuta infinitamente hasta que ocurre un evento que dispara una acción, en este caso, ese evento son las horas programadas para realizar el procesamiento de las imágenes enviadas desde la estación de captura.

Para utilizar el automático de procesamiento, es necesario compilar la función `auto_process_images.m` del directorio `src`, mediante el botón correspondiente en la interfaz. Sin embargo, si se quiere compilar manualmente, se ejecutan los siguientes comandos en la línea de comandos de MATLAB (suponiendo que el directorio de HORUS es `C:\horus`):

```
cd C:\horus
addpath('src')
mcc -m auto_process_images -a ./data -I ./tmp
```

Si la compilación se realizó mediante la interfaz, se genera el archivo ejecutable `run_auto_process_images.bat`. Al presionar el botón que lleva a cabo la compilación, se le pide al usuario ingresar el tamaño de paso y el error (en minutos) para la búsqueda de las imágenes que se desean procesar.

Si el archivo se compiló manualmente, en el sistema operativo Windows, se genera un ejecutable `auto_process_images.exe`, el cual recibe unos parámetros de entrada: nombre de la estación, tamaño de paso para la búsqueda de imágenes (en minutos) y error en la búsqueda (en minutos). Para ejecutarlo, se debe crear un archivo BAT que recibe los parámetros. Un ejemplo de archivo BAT para el automático de procesamiento podría contener el siguiente código (`run_auto_process_images.bat`):

```
auto_process_images.exe CARTAGENA 30 5
```

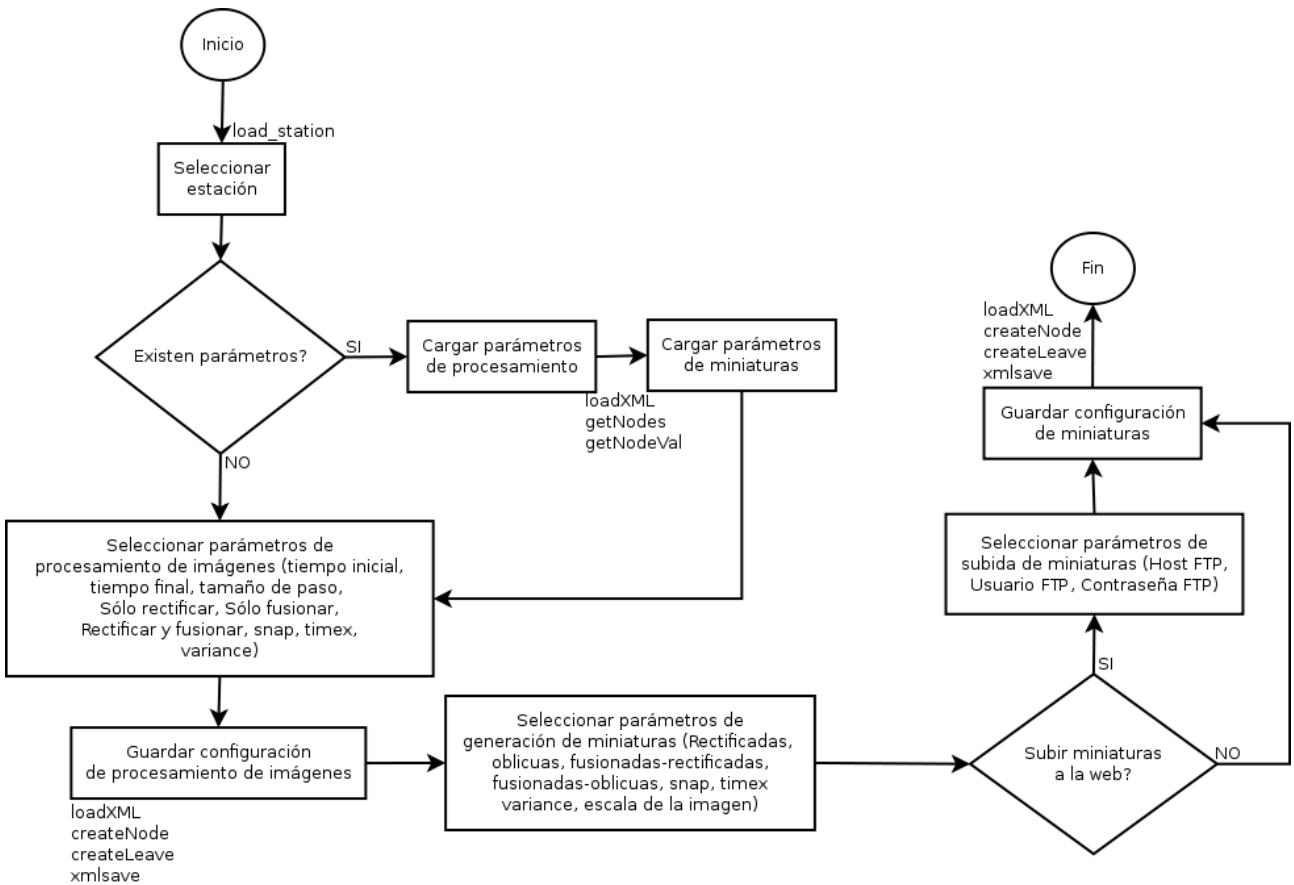


Figura 5.46: Diagrama de flujo del proceso de configuración de procesamiento

## 5.13. Configure database synchronization automatic

El sistema HORUS necesita una base de datos para almacenar y consultar toda la información necesaria para funcionar correctamente. Si hay muchas estaciones en diferentes lugares, lo deseable es tener una base de datos local para cada estación. Sin embargo, en ocasiones es necesario tener toda la información de las diferentes bases de datos locales o satélites, en una gran base de datos central que las reúna a todas. Por ejemplo, si se tiene un sitio web central, donde se puede ver la información de todas las estaciones, es necesario tener la información de todas las bases de datos disponibles.

Una manera de tener toda esta información en la base de datos central es ejecutar un programa de sincronización en cada estación para que a determinadas horas del día se sincronice el contenido de ambas bases de datos (la local y la central).

En esta sección, se describe la interfaz gráfica para configurar el automático de sincronización, el cual se encarga de ejecutar a determinadas horas del día el script `sync_databases.py` escrito en el lenguaje *Python*, al que se le pasan como parámetros los datos de la base de datos local y la remota, y realiza la sincronización del contenido. Todos los cambios se realizan sobre la base de datos central (remota).

La configuración realizada se almacena en el archivo `sync_info.xml` en el directorio `data`.

En la figura 5.48 se muestra la interfaz para configurar el automático de sincronización.

El comando para ejecutar esta interfaz es: `gui_configure_sync`.

Las componentes numeradas en la figura se explican a continuación:

1. Lista donde se escoge la estación.
2. Tiempo inicial del período de sincronización.

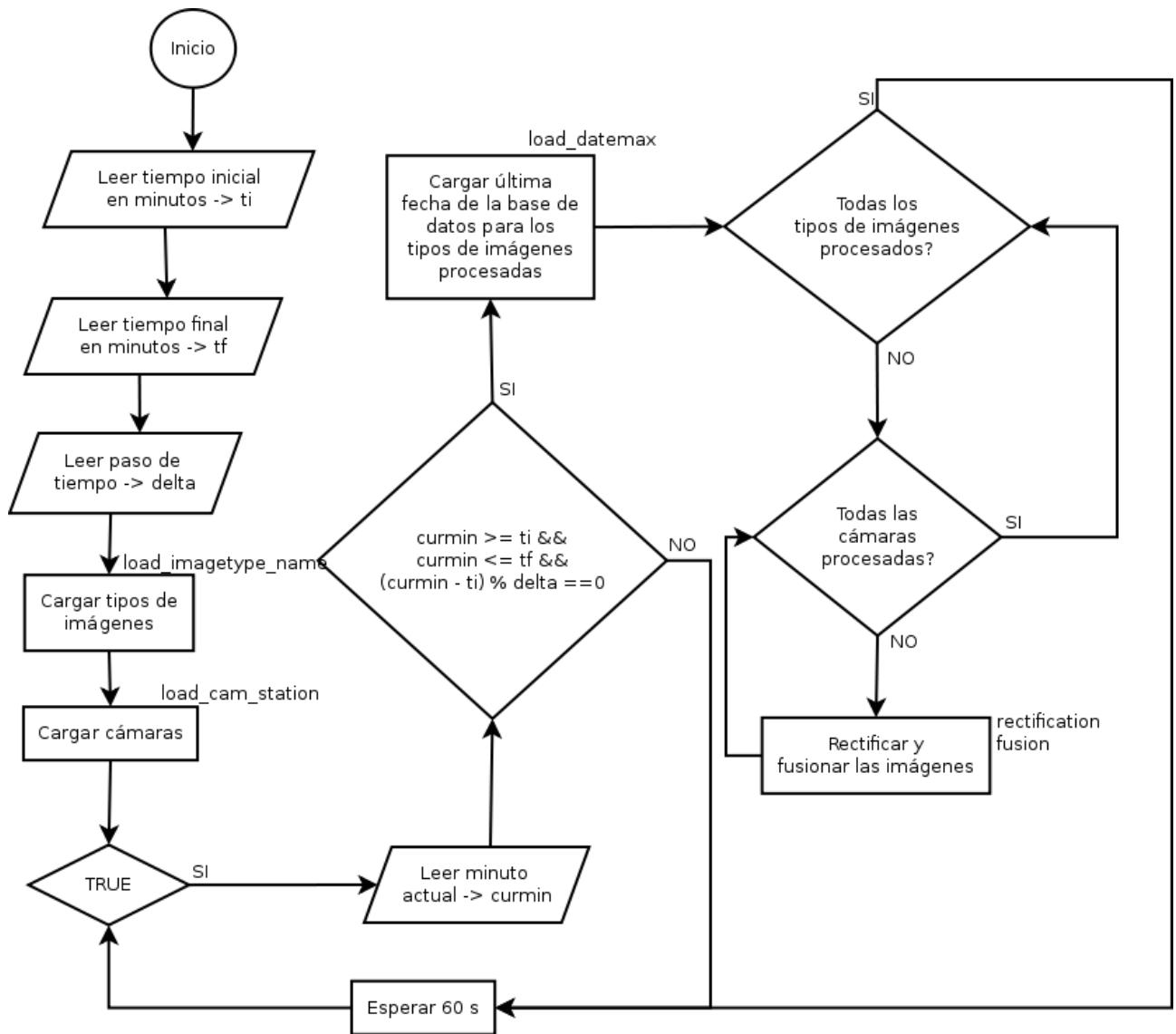


Figura 5.47: Diagrama de flujo del automático de procesamiento

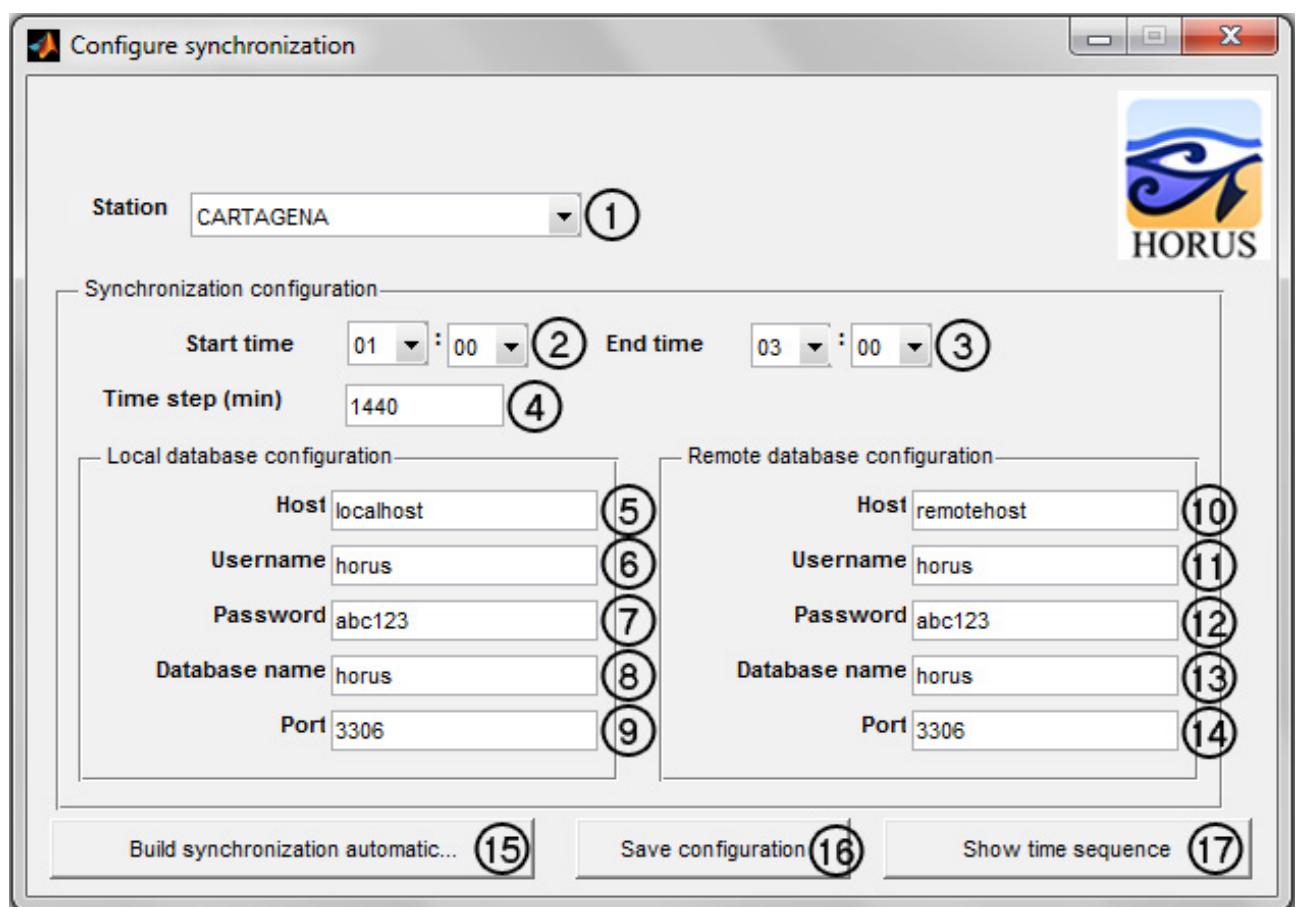


Figura 5.48: Interfaz para configurar el automático de sincronización

3. Tiempo final del período de sincronización.
4. Paso de tiempo, que indica cada cuánto se ejecuta la sincronización.
5. Dirección IP o nombre del *host* donde se encuentra la base de datos local.
6. Usuario de la base de datos local.
7. Contraseña de la base de datos local (al guardarla en el archivo XML, se encripta).
8. Nombre de la base de datos local.
9. Puerto donde escucha el servidor de la base de datos local.
10. Dirección IP o nombre del *host* donde se encuentra la base de datos remota.
11. Usuario de la base de datos remota.
12. Contraseña de la base de datos remota (al guardarla en el archivo XML, se encripta).
13. Nombre de la base de datos remota.
14. Puerto donde escucha el servidor de la base de datos remota.
15. Botón para generar el automático de sincronización.
16. Botón para guardar la configuración en el archivo `sync_info.xml`.
17. Botón para mostrar gráficamente los tiempos de ejecución de los automáticos de captura, transferencia, procesamiento y sincronización.

Para utilizar el automático de sincronización, es necesario compilar la función `auto_synchronization.m` del directorio `src`, mediante el botón correspondiente en la interfaz. Sin embargo, si se quiere compilar manualmente, se ejecutan los siguientes comandos en la línea de comandos de MATLAB (suponiendo que el directorio de HORUS es `C:\horus`):

```
cd C:\horus
addpath('src')
mcc -m auto_synchronization -a ./data -a ./src -I ./tmp
```

Si la compilación se realizó mediante la interfaz, se genera el archivo ejecutable `run_auto_synchronization.bat`.

Si el archivo se compiló en el sistema operativo Windows, se genera un ejecutable `auto_synchronization.exe`, el cual recibe como parámetro de entrada el nombre de la estación. Para ejecutarlo, se debe crear un archivo BAT que recibe el parámetro. Un ejemplo de archivo BAT para el automático de sincronización podría contener el siguiente código (`run_auto_synchronization.bat`):

```
auto_synchronization.exe CARTAGENA
```

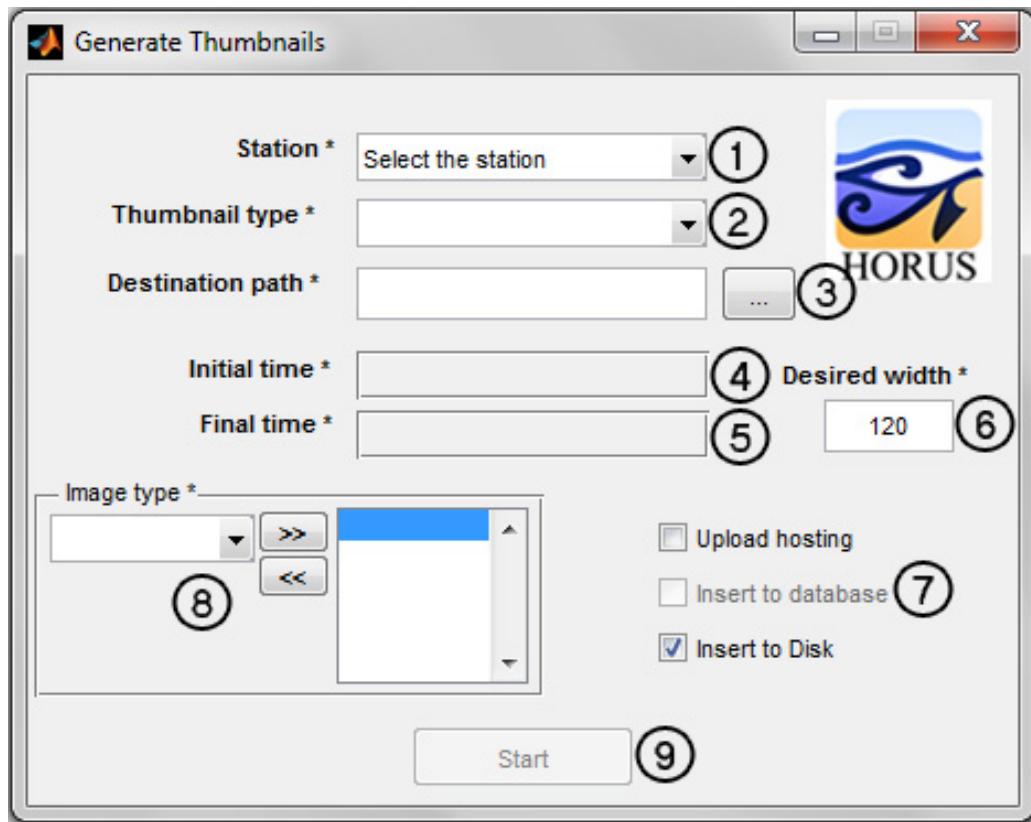


Figura 5.49: Interfaz inicial para generar miniaturas

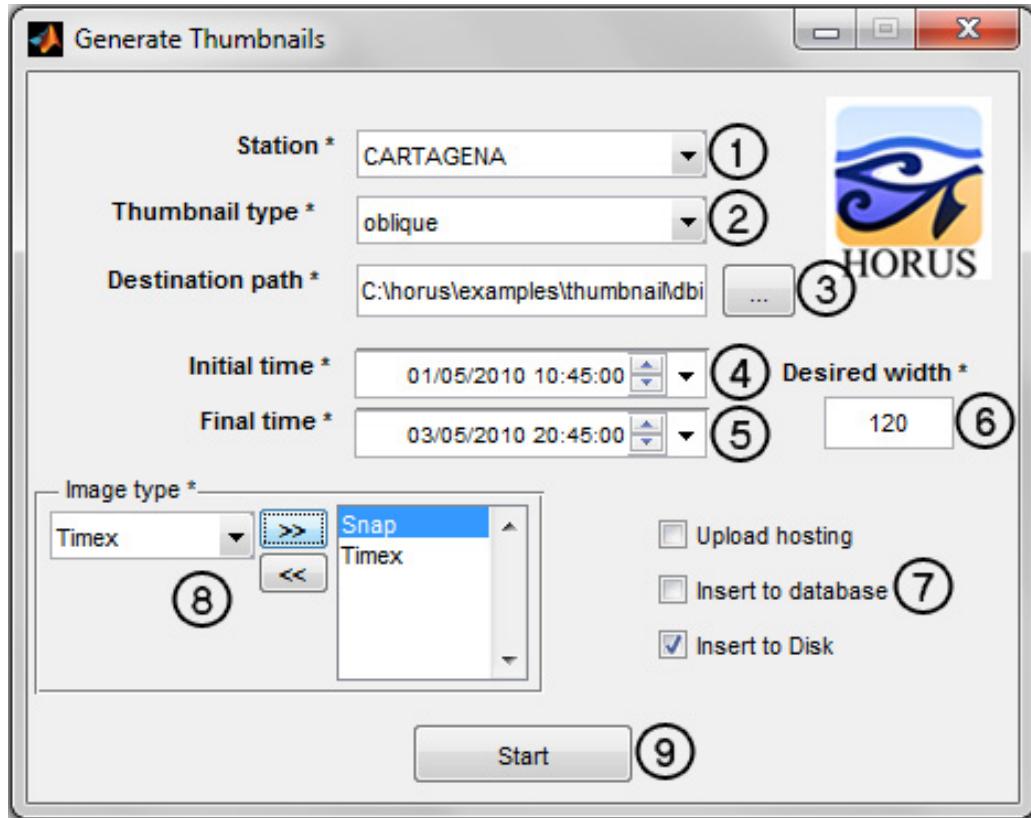


Figura 5.50: Interfaz con los datos necesarios para generar miniaturas

## 5.14. Configure thumbnail generation and upload

En el sistema HORUS se cuenta con la generación de imágenes miniaturas para mostrar una vista previa de las imágenes en la página web del sistema, [www.horusvideo.com](http://www.horusvideo.com). Con esta interfaz se podrán generar miniaturas dado el tipo de la miniaturización, el tipo de las imágenes y un rango de fechas. Con cada acción de modificación de la base de datos desde esta interfaz, se muestra una ventana indicando esto, la cual se cierra al terminar la operación. En las figuras 5.49 y 5.50 se muestra la interfaz.

El comando para ejecutar esta interfaz es: `gui_thumbnails_tool`.

Las componentes numeradas en la figura se explican a continuación (los campos con \* son obligatorios):

1. Lista para seleccionar la estación para la cual se quieren generar miniaturas.
2. Lista para seleccionar el tipo de las miniaturas que puede ser *oblique* para imágenes oblicuas, *rectified* para imágenes rectificadas, *merge\_oblique* para las imágenes oblicuas–rectificadas o *merge\_rectified* para las imágenes rectificadas–fusionadas.
3. Ruta donde se guardan las imágenes miniaturas, si se ha seleccionado en (7) la opción “*Insert to Disk*”, se debe seleccionar la ruta donde se van a guardar las imágenes, si se ha seleccionado la opción “*Insert to Database*”, aparecerá automáticamente la ruta en donde se guardarán las imágenes y no se podrá editar a menos que lo haga usando la interfaz para configurar las rutas. Si es el caso, la ruta se puede cambiar ya sea escribiendo la nueva ruta en este campo o haciendo clic en el botón ... que abrirá una nueva ventana donde se selecciona la ruta.
4. Fecha de inicio para generar las miniaturas.
5. Fecha final para generar las miniaturas.

Los campos (4) y (5) se mostrarán cuando se haya escogido un tipo en el campo (2) y se cambia la fecha moviendo los controles del calendario. Se debe tener en cuenta que la fecha final debe ser mayor que la de inicio.

6. Ancho en pixeles de las miniaturas. Se recomienda que el ancho de la miniatura sea aproximadamente 120 píxeles.
7. Se debe seleccionar si se desea que las imágenes sean subidas a un servidor web o si se desea que se guarde la información en la base de datos de HORUS y en la ruta configurada con anterioridad, o si desea que las imágenes se guarden sólo en el disco. Las opciones “*Insert to Disk*” y “*Insert to Database*” y también las opciones “*Insert to Disk*” y “*Upload hosting*” son excluyentes. Solo se podrán escoger en simultáneo las opciones “*Insert to Database*” y “*Upload hosting*”. Al seleccionar “*Upload hosting*” se creará un archivo con la información para subir las imágenes al servidor web. Para configurar los datos de conexión del *hosting* y subir las imágenes como tal se debe usar la interfaz `gui_upload_hosting`.
8. Selección de al menos un tipo de imagen para generar las miniaturas, los básicos son: *Snap*, *Timex* y *Var*. Los tipos de imágenes son leídos de la base de datos.
9. Botón para comenzar la miniaturización. Este botón se habilitará cuando se hayan ingresado los datos correctamente. Se comprueba que la ruta en (3) sea un directorio existente, si es una dirección web se mostrará vacío ya que estas rutas son de sólo lectura y se debe seleccionar una ruta correcta que esté en el disco duro. También se comprueba que se seleccione al menos un tipo de imagen y que se seleccione una de las opciones “*Insert to*

*Database*” o “*Insert to Disk*”. Este botón se deshabilitará mientras se generan las miniaturas y se abrirá una ventana de progreso en la cual se puede dar en cualquier momento clic en “*Cancel*”, para finalizar la operación. Al terminar se mostrará una ventana indicando el éxito o el fracaso de la operación.

En la figura 5.51 se muestra la interfaz que tiene como fin subir las miniaturas que se han generado, donde se ingresan los datos de conexión del servidor de datos. Las componentes numeradas en la figura se explican a continuación (Todos los campos son obligatorios):

1. Lista para seleccionar la estación para la que se quieren subir las imágenes miniaturas.
2. Host del servidor que puede ser una dirección IP o un nombre de dominio.
3. Se debe ingresar el usuario.
4. Se debe ingresar la contraseña.
5. Este botón es usado para guardar los datos de conexión. Sólo se habilita cuando se hayan diligenciado todos los campos.
6. Este botón inicia la subida de las imágenes al hosting. Se habilita después de guardar la configuración o al cargar una configuración existente.

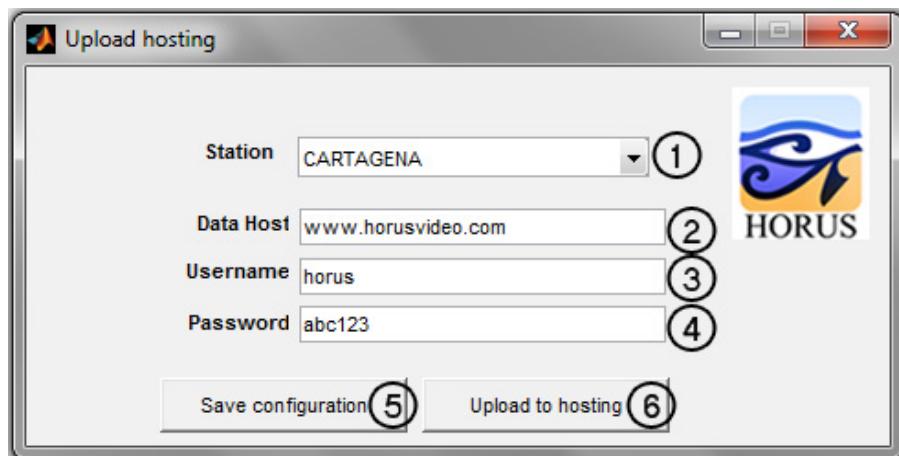


Figura 5.51: Interfaz para subir las miniaturas generadas a la web

Los datos de conexión son proporcionados por el equipo HORUS.

En la figura 5.52 se muestra un diagrama de flujo que explica los procesos de la interfaz donde para cada proceso se nombra la función que utiliza.

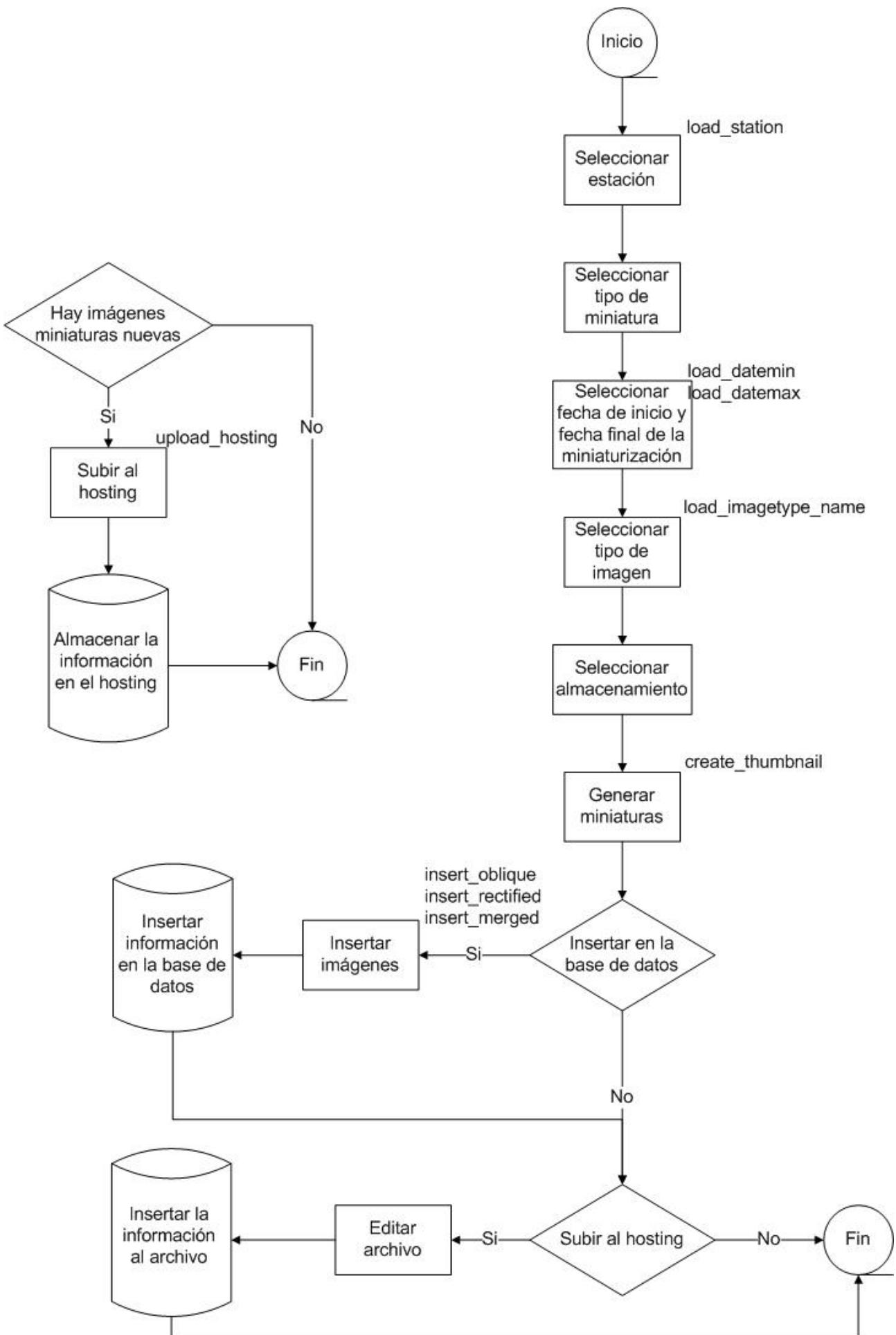


Figura 5.52: Diagrama de flujo interfaz Thumbnails Tool

# Capítulo 6

## FAQ

### 6.1. ¿Dónde puedo obtener más información sobre el Proyecto HORUS?

El Proyecto HORUS es desarrollado por la Universidad Nacional de Colombia Sede Medellín y la Universidad de Cantabria, y es financiado por la AECID (Agencia Española de Cooperación Internacional para el Desarrollo). En la página web <http://www.horusvideo.com> puede obtener más información sobre el Proyecto y las estaciones monitorizadas de Cartagena de Indias, Colombia, entre otros. Además, hay información de contacto y foros de discusión.

### 6.2. ¿Cómo puedo contactar al equipo HORUS?

A través de la página web <http://www.horusvideo.com>, o al correo de contacto [horus.unal@gmail.com](mailto:horus.unal@gmail.com).

### 6.3. ¿Cómo puedo acceder a la información de las estaciones monitorizadas por el Proyecto HORUS?

La información referente a las imágenes es libre y se puede descargar desde la web <http://www.horusvideo.com>. Su uso es autorizado por escrito previa solicitud al mail de contacto. La base de datos de las estaciones monitorizadas por el proyecto HORUS es restringida para los usuarios que tengan acceso autorizado. Sin embargo, es posible con el Toolbox de HORUS crear su propia base de datos para manejar su información.

### 6.4. ¿Qué requerimientos necesito para ejecutar las rutinas del Toolbox?

El Toolbox fue desarrollado en MATLAB R2011b, por lo tanto se recomienda que la versión de MATLAB que se utilice sea igual o mayor. Además, se necesitan los Toolboxes de MATLAB de Procesamiento de Imágenes (Image Processing), Adquisición de Imágenes (Image Acquisition) y Bases de Datos (Database). Además, para la conexión con la base de datos, se requiere el conector de JDBC para Java, y MySQL Server en caso de requerir configurar el servidor para la base de datos (ver capítulo 4).

## **6.5. ¿En qué sistema operativo funciona el Toolbox?**

El Toolbox fue diseñado para funcionar en Microsoft Windows y UNIX (Mac, Linux). Hay algunas diferencias entre los sistemas operativos, tales como el formato de las rutas y la compresión del video.

## **6.6. ¿Dónde puedo obtener documentación de las rutinas?**

En el paquete hay un directorio llamado `doc`, el cual contiene los manuales de usuario y técnicos. Además, cada función tiene un encabezado que describe su funcionamiento. En la línea de comandos de MATLAB puede ejecutar `help <función>` o `doc <función>` para acceder a esta ayuda.

## **6.7. ¿Puedo integrar mis desarrollos al paquete HORUS?**

Sí. El Proyecto HORUS es de naturaleza investigativa, y cualquier aporte en investigación que se haga es totalmente bienvenido.

## **6.8. ¿Qué limitaciones tiene el uso de este toolbox?**

El Toolbox de HORUS está en continuo desarrollo, por lo que muchas funcionalidades que se podrían incluir en el paquete aún no están implementadas. En cuanto a la utilización del paquete, se debe tener conocimientos en el lenguaje de MATLAB, y se debe tener la versión R2011b o superior para garantizar el correcto funcionamiento. Este toolbox no es de uso comercial y su uso debe hacerse con autorización expresa de los autores. La última versión de este software está disponible en la página web <http://www.horusvideo.com/horus.zip>.

## **6.9. ¿Ustedes diseñan e instalan estaciones?**

Sí. Se debe contactar por escrito al equipo HORUS enviando un correo electrónico a la dirección de contacto.

## **6.10. ¿Si yo tengo una estación la puedo integrar al portal HORUS?**

Sí. Se debe contactar por escrito al equipo HORUS enviando un correo electrónico a la dirección de contacto. Para integrar una nueva estación se debe incluir en la base de datos principal de HORUS y ésta al sitio web donde se sincronizan las imágenes.

## **6.11. ¿Quién diseñó el sistema?**

El sistema fue diseñado en un proyecto conjunto entre la Universidad de Cantabria, España y la Universidad Nacional de Colombia Sede Medellín, financiado por la Agencia Española de Cooperación Internacional para el Desarrollo (AECID).

## **6.12. ¿En el marco de qué proyecto se desarrolló esta versión y quien lo financió?**

Este sistema se desarrolló en el proyecto de transferencia tecnológica entre España y Colombia financiado por la AECID, llamado “Aplicación de Metodologías y Técnicas basadas en Sistemas de Vídeo para el Seguimiento de los Problemas Ambientales Costeros. Caso Cartagena, Colombia”.

# Bibliografía

- [1] Faugeras, O. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts. 2nd Edition. 1993.
- [2] Hartley R., Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York. 2nd Edition. 2003.
- [3] Salvi J., Armangué, X., Batlle, J. *A comparative review of camera calibrating methods with accuracy evaluation*. The Journal of Pattern Recognition. 35. pp. 1617–1635. 2002.
- [4] Wolf, P., DeWitt, B., Wilkinson, B. *Elements of Photogrammetry with Application in GIS*. McGraw–Hill, Massachusetts. 3nd Edition. 2000.
- [5] Holland, K. T., Holman, R. A., Lippmann, T. C., Stanley J., Plant, N. *Practical use of video imagery in nearshore oceanographic field studies*. IEEE Journal of Oceanic Engineering. 22. 1. pp. 81-92. 1997.
- [6] Osorio, A., Pérez, J. C., Ortiz, C., Medina, R. *Técnicas basadas en imágenes de video para cuantificar variables ambientales en zonas costeras*. Avances en Recursos Hídricos. 16. pp. 51-64. 2007.
- [7] Pérez, J. C. *Optimización No Lineal y Calibración de Cámaras Fotográficas*. Tesis de Maestría. Universidad Nacional de Colombia. 2009.