

▪ Proyecto Fin de Carrera ▪

Procesamiento de Imágenes Satelitales Multiespectrales

Ángel Javier Mena Espinosa

Junio 2014

Directores UPV/EHU:

Dr. Basilio Sierra Araujo

Dra. María del Carmen Hernández Gómez

Agradecimientos

Primero de todo quisiera agradecer a mis padres, Ángel y M^a Pilar, por todo lo que han hecho por mí (y lo que aún no han hecho) que me han dado una buena educación y unos buenos estudios. También quisiera agradecer al resto de mi familia, que siempre me preguntan cómo van los estudios.

Por otra parte quiero agradecer a mis directores de proyecto, Carmen y Basilio, por todo lo que me han ayudado a que este proyecto pudiera ser posible.

También quería expresar mi gratitud a toda la gente que me ha apoyado, no solo durante el proyecto, sino durante toda la carrera. Compañeros, amigos, profesores,... los cuales de una forma u otra me han ayudado o apoyado.

Resumen

En este proyecto se ha realizado el procesamiento de una imagen satelital multiespectral de México concretamente centrada en la región del Lago de Chapala. Este proceso tiene como objetivo la distinción de tierra y agua mediante un proceso semi-automático utilizando distintos *software* o herramientas informáticas.

Dentro del proyecto podemos destacar ciertas fases u operaciones como el preprocesado realizado a la imagen satelital donde se han aplicado una serie de transformaciones, la aplicación de técnicas de clasificación supervisada mediante la realización de entrenamiento y testeado con regiones de interés extraídas de la imagen satelital para la obtención de clasificadores o la aplicación de estos clasificadores en la binarización de la imagen, obteniendo una imagen binaria donde un valor representa agua y otro tierra.

También podemos destacar el empleo de índices de agua y vegetación como una herramienta fundamental en la detección y en el análisis de cuerpos de agua. Éstos han marcado la calidad de los resultados obtenidos en el proyecto.

Palabras clave: teledetección, procesamiento de imagen, Landsat 8, cuerpos de agua y clasificación supervisada.

ÍNDICE

Introducción	11
Capítulo 1: Documento de objetivos del proyecto	13
1.1 Objetivos y motivación.....	13
1.2 Planificación.....	14
1.2.1 Planificación temporal inicial.....	15
1.2.2 Planificación temporal real.....	16
1.3 Análisis de riesgos.....	18
1.3.1 Valoración de los riesgos.....	18
Capítulo 2: Análisis de antecedentes	21
2.1 Landsat 8.....	21
2.1.1 Programa Landsat.....	21
2.1.2 Un poco de historia.....	22
2.1.3 Cronología Landsat.....	23
2.1.4 Especificaciones técnicas Landsat 8.....	23
2.1.5 Bandas Landsat 7 y Landsat 8.....	24
2.1.6 Descripción de productos.....	26
2.1.7 Productos L1T.....	26
2.2 Bandas.....	28
2.2.1 Banda 1 (0.433-0.453 μm ; Coastal/Aerosol).....	28
2.2.2 Banda 2 (0.450–0.515 μm ; Blue).....	29
2.2.3 Banda 3 (0.525–0.600 μm ; Green).....	29
2.2.4 Banda 4 (0.433-0.453 μm ; Red).....	29
2.2.5 Banda 5 (0.845–0.885 μm ; NIR).....	30
2.2.6 Banda 6 (1.560–1.660 μm ; SWIR 1).....	30
2.2.7 Banda 7 (2.100–2.300 μm ; SWIR 2).....	31
2.2.8 Banda 8 (0.500–0.680 μm ; Panchromatic).....	31
2.2.9 Banda 9 (1.360–1.390 μm ; Cirrus).....	31
2.2.10 Bandas 10 (10.6-11.2 μm ; TIR 1) y 11 (11.5-12.5 μm ; TIR 2).....	32
2.2.11 Banda BQA.....	32
2.3 Combinaciones de bandas.....	32
2.3.1 Combinaciones de tres bandas.....	32
2.3.2 Operaciones entre bandas.....	34
2.4 Técnicas de detección de cuerpos de agua.....	36
2.4.1 Density slicing.....	36
2.4.2 Umbral sobre una única banda.....	36

2.4.3	Umbral sobre varias bandas.....	37
2.4.4	Umbral sobre combinaciones de bandas.....	37
2.4.5	Algoritmos de clasificación.....	37
2.5	Software a utilizar.....	38
2.5.1	GRASS GIS.....	38
2.5.2	Quantum GIS.....	40
2.5.3	R (lenguaje de programación).....	40
2.5.4	Weka.....	41
2.6	Algoritmos de clasificación.....	42
2.6.1	J48.....	42
2.6.2	NB (Naive Bayes).....	42
2.6.3	KNN.....	43
Capítulo 3: Diseño	45
3.1	Inicio.....	45
3.1.1	Primeros pasos.....	45
3.1.2	Software a emplear.....	46
3.2	Lectura de datos iniciales.....	46
3.3	Tratar valores nulos.....	49
3.4	Transformación a reflectancias.....	50
3.5	Generación de índices de vegetación y agua.....	51
3.6	Generación de una sub-imagen.....	53
3.6.1	Exportar.....	54
3.7	Generación de los ROI desde Quantum GIS.....	54
3.8	Rasterización de los ROI.....	56
3.8.1	Obtención del archivo '.shp'.....	57
3.8.2	Obtención del archivo '.csv'.....	57
3.9	Generación de la base de datos.....	58
3.10	Entrenamiento de clasificadores.....	60
3.11	Testeo de clasificadores.....	61
3.12	Clasificación.....	64
3.12.1	Creación de teselas.....	64
3.12.2	Clasificación.....	66
3.12.3	Función auxiliar 1.....	68
3.12.4	Función auxiliar 2.....	68
3.12.5	Función auxiliar 3.....	69
3.12.6	Función auxiliar 4.....	69
3.13	Obtención de resultados.....	70
3.14	Comparación resultados y NDWI.....	71
3.14.1	Umbrales para NDWI.....	71
3.14.2	Binarización de NDWI.....	72
3.14.3	Resultados finales.....	72
Capítulo 4: Resultados	75

4.1	Resultados entrenamiento (2 clases).....	75
4.1.1	J48.....	76
4.1.2	NB.....	76
4.1.3	KNN.....	76
4.1.4	Conclusiones.....	76
4.2	Resultados entrenamiento (5 clases).....	77
4.2.1	J48.....	77
4.2.2	NB.....	77
4.2.3	KNN.....	78
4.2.4	Conclusiones.....	78
4.3	Resultados testeo.....	78
4.3.1	J48.....	79
4.3.2	NB.....	79
4.3.3	KNN.....	79
4.3.4	Conclusiones.....	80
4.4	Resultados clasificación.....	80
4.4.1	J48.....	80
4.4.2	NB.....	81
4.4.3	KNN.....	81
4.4.4	Conclusiones.....	82
4.5	Comparación resultados y NDWI.....	83
4.5.1	J48.....	84
4.5.2	NB.....	84
4.5.3	KNN.....	85
4.5.4	Conclusiones.....	85
Capítulo 5: Conclusiones y líneas futuras		87
Capítulo 6: Bibliografía		89

ÍNDICE DE FIGURAS

Figura 1: Diagrama de Gantt inicial.....	16
Figura 2: Diagrama de Gantt real.....	17
Figura 3: Landsat 8 en órbita. Fuente: NASA y OSC.....	23
Figura 4: Comparativa de las bandas de Landsat 7 y Landsat 8. Fuente: USGS.....	25
Figura 5: Banda NIR.....	30
Figura 6: Combinación RGB en falso color.....	33
Figura 7: Índice NDVI.....	35
Figura 8: Ventana principal de GRASS.....	39
Figura 9: Ventana de visualización de GRASS.....	39
Figura 10: Pantalla de inicio de Weka.....	41
Figura 11: Pantalla de inicio de GRASS.....	47
Figura 12: Importar bandas en GRASS.....	48
Figura 13: Carga de bandas en GRASS.....	49
Figura 14: Propiedades conversión a reflectancias.....	50
Figura 15: Calculadora ráster de GRASS.....	52
Figura 16: Recorte a aplicar a la imagen.....	53
Figura 17: Bandas a utilizar en la creación de ROI en QGIS.....	55
Figura 18: Regiones de interés en QGIS.....	56
Figura 19: Conjunto de entrenamiento de banda SWIR2 en ancho de banda.....	64
Figura 20: Mosaico creado a partir de teselas en QGIS.....	71
Figura 21: Resultados J48.....	81
Figura 22: Resultados NB.....	81
Figura 23: Resultados KNN.....	82
Figura 24: Combinación RGB tierra/agua.....	82

ÍNDICE DE TABLAS

Tabla 1: Horas estimadas en el proyecto.....	15
Tabla 2: Horas empleadas en el proyecto.....	17
Tabla 3: Probabilidad e impacto de riesgos.....	19
Tabla 4: Comparativa de las bandas de Landsat 7 y 8.....	25
Tabla 5: Bandas de Landsat 8.....	28
Tabla 6: Combinaciones comunes de tres bandas en Landsat 8.....	33
Tabla 7: Operaciones más importantes entre dos bandas en Landsat 8.....	34
Tabla 8: Resultados 'cross-validation' para 2 clases.....	75
Tabla 9: Matriz de confusión de J48 para 2 clases.....	76
Tabla 10: Matriz de confusión de NB para 2 clases.....	76
Tabla 11: Matriz de confusión de KNN para 2 clases.....	76
Tabla 12: Resultados 'cross-validation' para 5 clases.....	77
Tabla 13: Matriz de confusión de J48 para 5 clases.....	77
Tabla 14: Matriz de confusión de NB para 5 clases.....	78
Tabla 15: Matriz de confusión de KNN para 5 clases.....	78
Tabla 16: Resultados del testeo.....	79
Tabla 17: Matriz de confusión de J48 en el testeo.....	79
Tabla 18: Matriz de confusión de NB en el testeo.....	79
Tabla 19: Matriz de confusión de KNN en el testeo.....	79
Tabla 20: Resultados clasificación.....	80
Tabla 21: Comparación resultados J48 y NDWI.....	84
Tabla 22: Comparación resultados NB y NDWI.....	84
Tabla 23: Comparación resultados KNN y NDWI.....	85

ÍNDICE DE ALGORITMOS

Algoritmo 1: Programa principal.....	59
Algoritmo 2: Primera parte del entrenamiento.....	61
Algoritmo 3: Preparación del conjunto de datos de test.....	62
Algoritmo 4: Entrenamiento de los clasificadores finales y realización del test.....	63
Algoritmo 5: Llamada de GRASS a R.....	65
Algoritmo 6: Creación de las teselas.....	65
Algoritmo 7: Llamada de GRASS a R.....	66
Algoritmo 8: Clasificación final.....	68
Algoritmo 9: Extracción de la información de las teselas.....	68
Algoritmo 10: Creación de máscara.....	69
Algoritmo 11: Extracción de información de máscara.....	69
Algoritmo 12: Clasificación de tesela.....	70

Introducción

Hoy en día, los beneficios otorgados por la tecnología han hecho que ésta sea cada vez más imprescindible en nuestra vidas. Una de estas tecnologías son los satélites. Éstos proporcionan gran cantidad de beneficios a nuestras vidas que muchas veces no tenemos conocimiento siquiera. De la misma forma que ignoramos ser observados desde arriba, la realidad es que somos constantemente observados por los satélites, concretamente, por la serie de satélites Landsat.

Los satélites de la serie Landsat llevan más de cuatro décadas observando la tierra. Han estado contemplando cómo ha cambiado el mundo durante los últimos 40 años. Mientras observaban la Tierra nos han mostrado y proporcionado imágenes de todo del globo terrestre, actualmente la serie Landsat cuenta con dos satélites activos en órbita: el satélite Landsat 7 y el satélite Landsat 8.

Landsat 8 es actualmente el último satélite de la serie Landsat. Fue lanzado el 11 de febrero de 2013 y declarado operativo el 30 de mayo del mismo año. Durante este periodo en órbita ha proporcionado imágenes de gran calidad de la tierra. Mediante sus dos sensores es capaz de captar imágenes desde el espectro visible hasta el infrarrojo térmico gracias a 10 bandas espectrales, abarcando las partes más importantes del espectro electromagnético.

Utilizando las 10 bandas espectrales podemos realizar gran cantidad de análisis, desde el estudio de la vegetación, de los cuerpos de agua, de las urbes, distinción de regiones de interés, ... Para facilitarnos esta tarea, a lo largo de los años y gracias a la evolución de la tecnología, han

surgido diferentes *software*, conocidos como 'Sistemas de Información Geográfica' (SIG) o 'Geographic Information System' (GIS). Estos están diseñados para poder realizar gran cantidad de tareas del campo geográfico, desde capturar, almacenar, manipular, analizar y hasta desplegar información.

Dentro de estos *software* GIS se encuentran Quantum GIS y GRASS GIS, dos *software* libres especializados en tareas de procesado de imágenes multiespectrales que permiten desde la manipulación de valores hasta la conversión de un tipo de datos a otro. También permiten crear y manipular regiones de interés (ROI) para diferentes fines.

No siempre todas las tareas se pueden realizar empleando una o dos herramientas. Generalmente a la hora de hacer cierto trabajo, empleamos varias herramientas y cada una para una tarea diversa. Actualmente existen herramientas que permiten realizar múltiples tareas. Una herramienta de este tipo es el lenguaje de programación R. Este es un lenguaje y entorno de programación para análisis estadístico y gráfico, mediante el cual podemos realizar tareas de clasificación: creación de bases de datos, entrenamiento de clasificadores, clasificación supervisada y no supervisada,... pudiendo emplear los clasificadores más importantes dentro del mundo del aprendizaje automático y de la minería de datos.

Uniendo el lenguaje de programación R y los Sistemas de Información Geográfica, Quantum GIS y GRASS GIS, podemos alcanzar el objetivo final del proyecto: la distinción de agua y tierra de una imagen satelital multiespectral perteneciente al satélite Landsat 8. Para satisfacer este objetivo necesitaremos adquirir una serie de conocimientos, para lo cual estudiaremos el estado del arte, profundizaremos en el manejo del *software* y realizaremos los programas necesarios.

'1'

Capítulo 1: Documento de objetivos del proyecto

1.1 OBJETIVOS Y MOTIVACIÓN

El objetivo principal de este proyecto es la distinción de agua y tierra de una imagen satelital multiespectral de México centrada en la región del lago de Chapala. Se pretende hacer un estudio completo de los requisitos necesarios para esta tarea, desde el preprocesado de la imagen hasta su clasificación final. Mediante este proyecto se pretende abrir la puerta a futuras investigaciones donde se emplee varias imágenes pertenecientes a la misma región en diferentes épocas del año o a imágenes de diferentes regiones, utilizando el nuevo satélite de la serie Landsat, el número 8.

Dentro de los objetivos se encuentra el estudio y manejo de las herramientas empleadas para esta tarea: herramientas GIS, concretamente Quantum GIS y GRASS GIS y el lenguaje de programación R aplicado al uso de procesamiento de imágenes satelitales.

1.2 PLANIFICACIÓN

El proyecto consistirá en una serie de tareas o pasos a realizar. Dado que nos enfrentamos a un proyecto de investigación con un final difuso, no está claro cuál será el límite al que se llegue. Por lo tanto planificaremos el proyecto a 'alto nivel' mediante pocas tareas, de gran volumen, que engloben a tareas más pequeñas y nos abstraigan de éstas. A continuación, presentamos las tareas planificadas:

- **Estudio del arte:** El alumno se familiarizará brevemente con el procesamiento de imágenes satelitales, las principales herramientas y los métodos empleados en este área.
- **Aprendizaje organizativo:** Se obtendrán los conocimientos necesarios para manejar correctamente las herramientas a utilizar (GRASS GIS, Quantum GIS, lenguaje R,...), así como los conocimientos del campo del procesamiento de imágenes satelitales.
- **Desarrollo técnico:** Consta de las siguientes fases:
 - **Preprocesado en GRASS GIS:** Este consistirá en aplicar una serie de procesos a la imagen inicial para obtener la imagen (varias bandas) con la cual se trabajará posteriormente para obtener los resultados finales. En este preprocesado se realizarán las siguientes tareas: tratamiento de valores nulos, transformación a reflectancias y generación de índices de agua.
 - **Generación de regiones de interés (ROI):** Mediante Quantum GIS se escogerán una serie de regiones, las cuales posteriormente nos servirán para la clasificación supervisada siendo los datos de entrenamiento de la clasificación. Si fuera necesario, se empleará GRASS GIS junto a Quantum GIS, para obtener las regiones de interés en el formato adecuado para la clasificación posterior.
 - **Creación de la base de datos:** Uniendo las bandas resultantes del preprocesado y las regiones de interés se creará la base de datos: las bandas y las regiones de interés. Este proceso y el resto de la clasificación se realizará en el lenguaje R.
 - **Obtención de clasificadores:** Se crearán varios clasificadores de entre los clasificadores más importantes dentro del aprendizaje automático o la minería de datos. Estos clasificadores serán entrenados con el conjunto de entrenamiento escogido previamente. También se realizará un testeo con otras regiones de interés a fin de confirmar la calidad de los clasificadores.
 - **Clasificación:** Empleando los clasificadores creados previamente se clasificará la imagen satelital multiespectral. De la clasificación obtendremos una imagen binaria por cada clasificador empleado, con valores 0 (agua) y 1 (tierra).

- **Análisis de los resultados:** Se analizarán los resultados obtenidos en la clasificación. Para valorar su calidad serán comparados con los índices de agua utilizados en el preprocesamiento.
- **Generación de la memoria:** Esta tarea será la encargada de plasmar todo el trabajo realizado en el proyecto, ha de ser realizada cuidadosamente, porque reflejará todo el trabajo realizado.

1.2.1 Planificación temporal inicial

Completaremos la planificación con la estimación de horas (tabla 1) y con su distribución en el tiempo mediante un diagrama de *Gantt* (figura 1).

Tarea principal	Subtarea	Horas
Estudio del arte		10
Aprendizaje organizativo		110
Desarrollo técnico		
	Preprocesado en GRASS	20
	Generación de ROI	20
	Creación de BD	30
	Obtención de clasificadores	30
	Clasificación	30
Análisis de resultados		20
Generación de la memoria		80
Total		350

Tabla 1: Horas estimadas en el proyecto.

Inicialmente, el alumno se introducirá en el área del proyecto. Dado el desconocimiento del alumno en muchas de las tareas a las cuales se enfrenta, necesitará emplear un gran número de horas en el aprendizaje de las distintas herramientas, para poder realizar las tareas posteriormente. Las tareas de desarrollo técnico se estiman con una duración de 20-30 horas y de forma similar con una duración de 20 horas se estima el análisis de resultados. Por último, se han previsto una gran cantidad de horas para la adecuada generación de la memoria.

En el diagrama de *Gantt* (figura 1) podemos observar cómo comenzaremos el proyecto con el estudio del arte. Dos semanas después empezaremos con el aprendizaje organizativo, el

desarrollo técnico y la generación de la memoria de forma simultánea y teniendo de duración la totalidad del proyecto (salvo el desarrollo técnico que finalizará tres semanas antes).

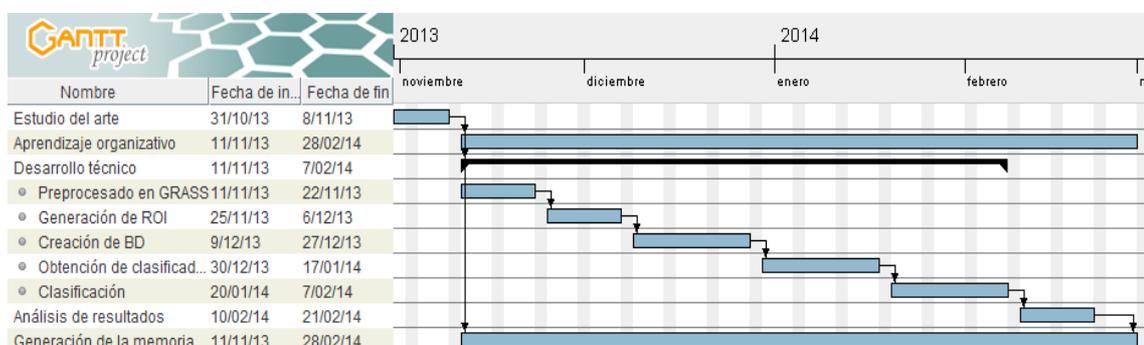


Figura 1: Diagrama de Gantt inicial.

Se ha decidido empezar de forma simultánea las fases de desarrollo y de aprendizaje organizativo a fin de obtener una rápida puesta en práctica de los conocimientos adquiridos. En el mismo momento, la generación de la memoria dará comienzo con el fin de documentar el desarrollo hecho a medida que se vaya realizando para una vez realizada una tarea sea documentada inmediatamente.

El resto de tareas (de desarrollo y análisis) serán realizadas de forma secuencial, teniendo dependencia de la tarea anterior. Sólo una vez terminada una tarea, se podrá pasar a la siguiente. Esto es debido a que estas tareas necesitan los resultados obtenidos de sus predecesoras.

1.2.2 Planificación temporal real

Inicialmente hicimos una planificación, pero ésta no pudo cumplirse lo que introdujo grandes cambios en la misma. Para observar mejor los cambios en la planificación empleando la misma tabla de horas (tabla 2), añadiremos un valor sumando o restando, indicando cuántas horas aumentó o disminuyó la duración de cada tarea.

Como observamos en la tabla 2, algunas tareas no variaron en duración o la variación en el número de horas fue pequeña. Estos desfases en la cuantificación horaria fueron debidos a la no correcta estimación de la duración de las tareas, o a la necesidad de repetir algunos pasos.

Se tuvieron que repetir algunos pasos en el preprocesado; al no haber realizado el proceso de tratamiento de valores nulos (apartado 3.3). También hubo problemas durante la creación de clasificadores. Inicialmente empleamos el paquete 'Rweka' para el entrenamiento de clasificadores (mediante el lenguaje de programación R), pero este paquete no realizaba la 'cross-validation' como se esperaba, obligándonos a tener que realizarla a mano. Se decidió cambiar al paquete 'Caret' para solucionar esta problemática.

Tarea principal	Subtarea	Horas
Estudio del arte		10
Aprendizaje organizativo		110 -10
Desarrollo técnico		
	Preprocesado en GRASS	20 +5
	Generación de ROI	20 +5
	Creación de BD	30
	Obtención de clasificadores	30 +10
	Clasificación	30 +25
Análisis de resultados		20 -5
Generación de la memoria		80 +10
Total		350 +40

Tabla 2: Horas empleadas en el proyecto.

Por otra parte, la tarea de clasificación varió considerablemente. Esto principalmente se debió a dos motivos. La clasificación no pudo hacerse directamente debido al gran volumen de datos ya que cierta parte del código era incapaz de trabajar con un volumen tan grande de datos. Por lo tanto se tuvo que buscar un método alternativo. El otro problema fueron los resultados obtenidos en la clasificación. Éstos contenían ruido en los bordes, ya que el algoritmo tomaba los valores nulos como valores a clasificar.

Pero las tareas no sólo fueron modificadas a nivel de duración, sino también en su distribución en el tiempo. Esto podemos observarlo en el diagrama de Gantt de la figura 2.

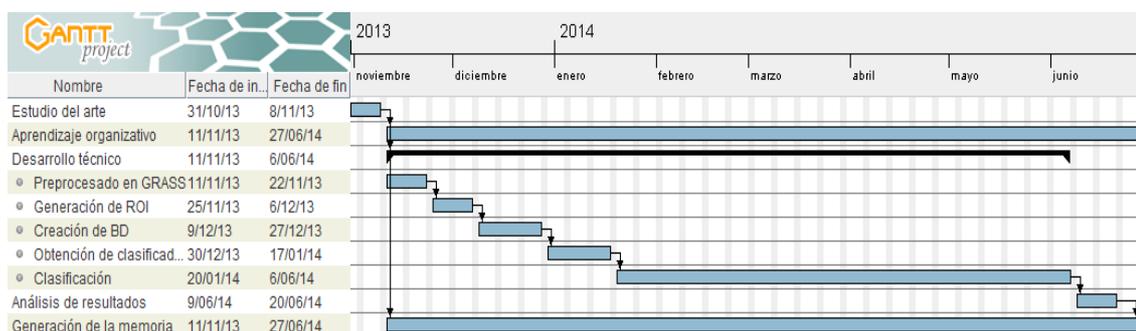


Figura 2: Diagrama de Gantt real.

El cambio se produjo en las tareas previstas para febrero y marzo, como puede observarse en el diagrama de Gantt, dichas tareas fueron prolongadas. El principal motivo no fue el aumento

de la duración de las tareas, si no la materialización de un riesgo, el 'sobre-exceso de trabajo'. Al comienzo del segundo cuatrimestre del curso académico, debido al exceso de carga lectiva, el alumno no pudo dedicar el tiempo requerido al proyecto tal y como estaba planificado.

1.3 ANÁLISIS DE RIESGOS

Para que el proyecto llegue a buen puerto es necesario el análisis de los posibles riesgos. También se tendrá un plan de contingencia en caso de que éstos se hagan realidad para solventarlos o minimizar sus efectos.

- **Pérdida de información:** La pérdida de parte o de toda la información podría ser grave para el proyecto. Dada la naturaleza del material de trabajo, al emplear gran cantidad de información debido al tamaño de los archivos de imágenes, se realizará una copia de seguridad completa cada dos semanas. Además, cada semana se realizará una copia de seguridad excluyendo los archivos de gran volumen. La copia de seguridad completa se realizará en una memoria externa mientras que la semanal se realizará en la nube y en otra memoria externa.
- **Problemas de *software* o *hardware*:** En el caso de tener problemas con alguna herramienta y si estos problemas fuesen graves y no se solucionasen, se realizará la reinstalación de la herramienta para lo cual se guardarán los instaladores de todas ellas. Además, la última versión de GRASS GIS (versión 7.0) se encuentra en estado de revisión (versión beta), por lo tanto se dispondrá de la versión estable anterior (la 6.4) como segunda opción.
- **Disponibilidad de los recursos:** En caso de la no disponibilidad de un recurso que impediría la realización de la tarea activa, se pasará a la siguiente tarea más urgente disponible mientras se espera a que el recurso vuelva a estar disponible.
- **Sobre-exceso de trabajo:** Por motivos laborales, académicos o personales, podría darse el caso que el alumno no disponga del tiempo necesario para trabajar en el proyecto como está planificado. En tal caso, las actividades se pospondrán o se ralentizarán. En la medida de lo posible esta situación será aceptada, siempre y cuando esté justificada.
- **Problemas de salud:** En caso de que se diera un problema de salud, por parte del alumno o de alguno de los directores y si alguna tarea se viera afectada, esta podría retrasarse, dado el gran margen de tiempo disponible.

1.3.1 Valoración de los riesgos

Mediante la siguiente tabla se valorarán los riesgos presentados, mediante probabilidad e impacto de estos:

	Probabilidad de ocurrir	Nivel de impacto
Pérdida de información	Baja	Alto
Problemas de software o hardware	Medio	Bajo
Disponibilidad de los recursos	Medio	Bajo
Sobreexceso de trabajo	Alto	Alto
Problemas de salud	Baja	Bajo

Tabla 3: Probabilidad e impacto de riesgos.

Esta tabla se tendrá en cuenta para valorar los riesgos así como los planes de contingencia para minimizar las consecuencias.

'2'

Capítulo 2: Análisis de antecedentes

2.1 LANDSAT 8

Landsat 8 (acrónimo de '*Land Satellite*') es un satélite de observación terrestre de la serie de satélites Landsat. También llamado LDCM, '*Landsat Data Continuity Mission*' (Misión de Continuidad de Datos de Landsat), es el octavo satélite de su serie como su nombre indica. Fue lanzado el 11 de febrero de 2013 desde la base aérea de Vandenberg en California y, posteriormente, fue declarado operativo el 30 de mayo del mismo año. Actualmente, sigue operativo y es el satélite con el que trabajaremos en este proyecto.

2.1.1 Programa Landsat

Los 'Landsat' son una serie de satélites construidos y puestos en órbita por EE. UU. para la observación en alta resolución de la superficie terrestre siendo la primera serie de sensores de adquisición de imágenes satelitales. El 23 de julio 1972 se puso en marcha el '*Earth Resources Technology Satellites*', el primer satélite de la serie, el cual después fue renombrado como Landsat.

A lo largo de los años, los instrumentos de los satélites Landsat han adquirido millones de imágenes. Estas imágenes han sido conservadas en EE. UU. y en estaciones Landsat repartidas por todo el mundo. Son un recurso único para la investigación del cambio del mundo además de tener grandes aplicaciones en la agricultura, la cartografía, la geología, la silvicultura, la ordenación del territorio, la vigilancia y la educación.

2.1.2 Un poco de historia

La empresa '*Hughes Aircraft*', en el centro de investigación Santa Barbara, inició el diseño y desarrollo de los tres primeros escáneres multispectrales en 1969. El primer prototipo MSS se terminó en el plazo de nueve meses, en 1970. Entonces se puso a prueba mediante el escaneo de '*Half Dome*' en el Parque Nacional de Yosemite.

El programa se llamó '*Earth Resources Technology Satellites*' cuando se inició en 1966, pero el nombre fue cambiado a Landsat en 1975. En 1979, la 'Directiva Presidencial 54' del presidente Jimmy Carter de EE. UU., transfirió las operaciones Landsat de la NASA ('*National Oceanic and Atmospheric Administration*'), a la NOAA ('*National Oceanic and Atmospheric Administration*').

En 1985, la empresa EOSAT ('*Earth Observation Satellite Company*'), una asociación de '*Hughes Aircraft*' y RCA, fue seleccionada por la NOAA para operar el sistema Landsat durante diez años. EOSAT trabajaba con los satélites Landsat 4 y 5, y además tenía los derechos para comercializar los datos de Landsat y para construir los satélites Landsat 6 y 7.

En 1989, el financiamiento de la NOAA para el programa Landsat se agotó. NOAA ordenó el cierre de Landsat 4 y 5. El vicepresidente Dan Quayle tomó nota de la situación y dispuso fondos de emergencia, los cuales permitieron que el programa continuara con los archivos de datos intactos.

En 1990 y 1991, el Congreso de Estados Unidos proporcionó sólo la mitad de la financiación del año a la NOAA y solicitó que los organismos que utilizaban datos de Landsat proporcionaran los fondos restantes para los otros seis meses del próximo año. En 1992, se hicieron varios esfuerzos para conseguir fondos para seguir con el programa Landsat pero, a finales del año, la EOSAT cesó el procesamiento de datos de Landsat. El 5 de octubre de 1993 fue lanzado Landsat 6, pero se perdió debido a un fallo de lanzamiento. Posteriormente, en 1994, el procesamiento de datos de Landsat 4 y 5 se reanudó por EOSAT. Finalmente, el 15 de abril de 1999, la NASA lanzó Landsat 7.

El valor del programa Landsat fue reconocido por el Congreso de Estados Unidos en octubre de 1992, cuando se aprobó la ley de '*Land Remote Sensing Policy Act*' la cual autorizó la adquisición de Landsat 7 y aseguró la disponibilidad permanente de los datos y de las imágenes digitales de Landsat, con el menor coste posible, para antiguos y nuevos usuarios.

2.1.3 Cronología Landsat

- Landsat 1: lanzado el 23 de julio de 1972 y finalizado el 06 de enero de 1978.
- Landsat 2: lanzado el 22 de enero de 1975 y finalizado el 22 de enero de 1981.
- Landsat 3: lanzado el 5 de marzo de 1978 y finalizado el 31 de marzo de 1983.
- Landsat 4: lanzado el 16 de julio de 1982 y finalizado el 14 de diciembre de 1993.
- Landsat 5: lanzado el 1 de marzo de 1984 y dado de baja el 5 de junio de 2013 siendo el satélite Landsat con el mayor ciclo de vida de todos.
- Landsat 6: lanzado el 5 de octubre de 1993. No pudo llegar a la órbita.
- Landsat 7: lanzado el 15 de abril de 1999 y aún en funcionamiento. Pero desde 2003 con la línea del corrector de exploración defectuosa (*gap fill problem*)¹.
- Landsat 8: lanzado el 11 de febrero de 2013 y actualmente operativo.



Figura 3: Landsat 8 en órbita. Fuente: NASA y OSC.

2.1.4 Especificaciones técnicas Landsat 8

¹ http://landsat.usgs.gov/using_Landsat_7_data.php

El satélite Landsat 8 dispone de dos instrumentos fundamentales:

- '*Operational Land Imager*' (OLI), sensor que capta el espectro visible, infrarrojo cercano (NIR) e infrarrojo de onda corta (SWIR) y tiene una resolución espacial de 15 metros en el canal pancromático y de 30 metros en los canales multiespectrales.
- '*Thermal Infrared*' Sensor (TIRS), sensor que mide la temperatura de la superficie terrestre mediante dos bandas del infrarrojo térmico (bandas 10 y 11). Gracias a los principios de la Física es capaz de distinguir la temperatura de la superficie de la tierra de la temperatura de la atmósfera. Los datos generados por este sensor tienen una resolución espacial de 100 metros y son de gran valor para medir la evapotranspiración y el consumo de agua en agricultura.

Los sensores OLI y TIRS escanean la superficie terrestre por líneas utilizando un sensor de tipo escoba, con una franja de 180 km y tienen una resolución radiométrica de 12 bits ($2^{12}=4096$).

Con estos sensores, el valor de un píxel está comprendido entre 0-4095 (4096 niveles potenciales), frente a los 256 niveles potenciales que se conseguían con una resolución de 8 bits ($2^8=256$). Esto permite distinguir con mayor facilidad las características que son espectralmente parecidas.

2.1.5 Bandas Landsat 7 y Landsat 8

Más adelante analizaremos en profundidad las distintas bandas de Landsat 8, pero antes hemos de analizar las similitudes y diferencias de las bandas de Landsat 7 y 8.

El satélite Landsat 8 dispone de dos tipos de sensores, OLI y TIRS mientras que Landsat 7, por su parte, emplea el sensor ETM+ ('*Enhanced Thematic Mapper Plus*').

Las bandas espectrales del sensor OLI, aunque similares a las bandas del sensor ETM+, tienen ciertas diferencias. La primera diferencia es que Landsat 8 tiene más bandas que Landsat 7. Se han incluido dos bandas nuevas: la banda 1 (*Costal/Aerosol*) y la banda 9 (*Cirrus*), que analizaremos más adelante.

Por otra parte, el sensor TIRS recoge dos bandas térmicas: las bandas 10 y 11 cuyas longitudes de onda cubren el rango de la banda 6 del sensor ETM+. Es decir, la antigua banda termal se ha dividido en dos.

El resto de bandas persisten, pero sus respectivos rangos espectrales han disminuido en mayor o menor medida. En la siguiente tabla se pueden apreciar las bandas de ambos satélites con su ancho de banda y su resolución.

Landsat 7			Landsat 8		
Banda	Ancho (μm)	Resolución (m)	Banda	Ancho (μm)	Resolución (m)
			1 Coastal	0.43 – 0.45	30
1 Blue	0.45 – 0.52	30	2 Blue	0.45 – 0.51	30
2 Green	0.52 – 0.60	30	3 Green	0.53 – 0.59	30
3 Red	0.63 – 0.69	30	4 Red	0.64 – 0.67	30
4 NIR	0.77 – 0.90	30	5 NIR	0.85 – 0.88	30
5 SWIR1	1.55 – 1.75	30	6 SWIR1	1.57 – 1.65	30
7 SWIR2	2.09 – 2.35	30	7 SWIR2	2.11 – 2.29	30
8 Pan	0.52 – 0.90	15	8 Pan	0.50 – 0.68	15
			9 Cirrus	1.36 – 1.38	30
6 TIR	10.40 – 12.50	30/60	10 TIRS1	10.6 – 11.19	100*
			11 TIRS2	11.5 – 12.51	100*

Tabla 4: Comparativa de las bandas de Landsat 7 y 8.

*Las bandas termales de Landsat 8 se obtienen con una resolución espacial de 100 metros, pero se remuestran a una resolución espacial de 30 metros.

Podemos comparar mejor las bandas mediante la gráfica de la figura 4. Esta tiene representadas las distintas bandas de ambos satélites con su rango espectral correspondiente.

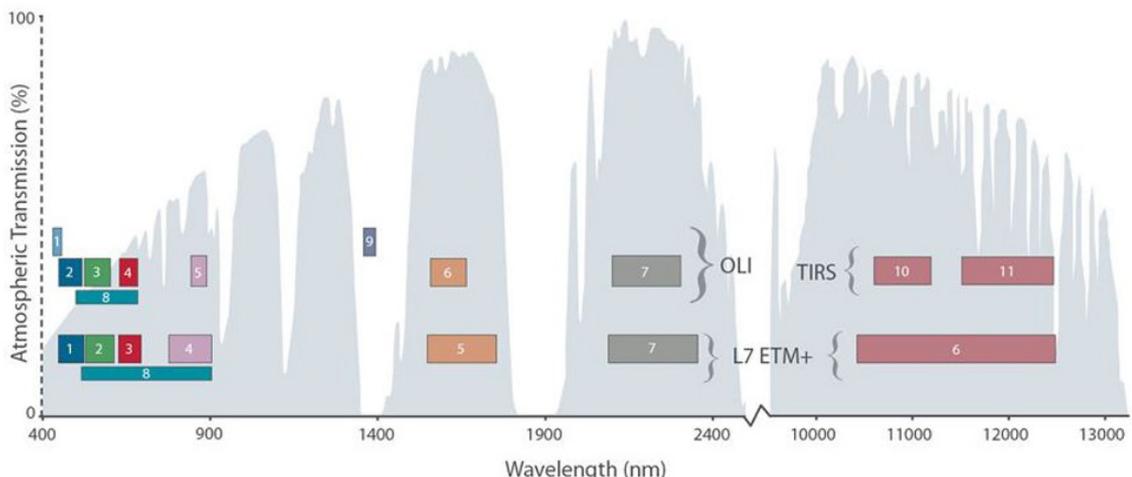


Figura 4: Comparativa de las bandas de Landsat 7 y Landsat 8. Fuente: USGS.

Adicionalmente, en cada producto Landsat 8, se incluye una banda de control de calidad: la banda BQA. Esta proporciona información más detallada sobre la presencia de características tales como nubes, agua y nieve.

2.1.6 Descripción de productos

La NASA realiza ciertos pre-procesos de las imágenes satelitales multiespectrales antes de ponerlas a disposición de los usuarios. Los usuarios finales obtienen productos L1Gt o L1T que están predefinidos en formato 'geoTIFF' ('*Geographical Tagged Image File Format*').

A continuación, se describen algunas de las definiciones de tipo de producto proporcionadas por el USGS ('*U.S. Geological Survey*'). Estas definiciones nos dan una idea de la nomenclatura utilizada y una referencia de las relaciones entre los tipos de productos:

- **Productos de Nivel 0 (L0):** Los productos del tipo L0 son imágenes digitales con todas las transmisiones de datos y objetos sin formato. Estos productos están al mismo tiempo ordenados de manera proporcional, espacial y secuencialmente por las bandas multiespectrales.
- **Productos de Nivel 1 Radiometric (L1R):** Los productos del tipo L1R consisten en datos de imágenes radiométricamente corregidas. Estos se derivan de datos L0 y son escalados a valores de radiancia espectral o reflectancia.
- **Productos de Nivel 1 Systematic (L1G):** Los productos del tipo L1G consisten en productos de datos del tipo L1R con correcciones geométricas sistemáticas aplicadas y muestreos para el registro en un sistema de proyección cartográfica. Estos datos se encuentran referenciados al Sistema Geodésico Mundial de 1984 (WGS84), G873.
- **Productos de Nivel 1 Gt (L1Gt):** Los productos del tipo L1Gt consisten en datos del tipo L1G con información de posición a bordo o efemérides definitivas, así como datos controlados de elevación para corregir los errores de posición.
- **Productos de Nivel 1 Terrain (L1T):** Los productos de datos L1T consisten en productos de datos L1R con correcciones geométricas sistemáticas aplicadas, utilizando para ello puntos de control terrestre (GCP) o información de posición integrada a bordo para entregar una imagen registrada a una proyección cartográfica, referenciada a WGS84, G873. Además, los datos también contienen una corrección topográfica por el desplazamiento del terreno debido al relieve.

2.1.7 Productos L1T

Los productos estándar L1T son productos que se encuentran en formato de niveles digitales enteros (DN) con una resolución radiométrica de 16 bits. Éstos se pueden convertir a valores de reflectancia TOA (*Top Of Atmosphere*) (bandas 1-9) o radiación (bandas 1-11) con factores de escala previstas en el archivo de metadatos incluido en el producto. Los datos de una imagen L1Gt o L1T se encuentran radiométrica y geoméricamente corregidos y están disponibles en formato 'geoTIFF'.

El producto final de salida es un archivo con extensión '.tar.gz'. Estos archivos se graban en un formato del tipo '.tar' y luego son comprimidos con la aplicación 'gzip'. Los archivos '.tar' no contienen ninguna información de los sub-directorios. Por lo tanto, se debe descomprimir todos los archivos directamente en el directorio actual.

- LC82220052014265LGN00.tar.gz

Este es un ejemplo de un producto final L1G para una determinada imagen que contiene los siguientes sub-archivos:

- LC82220052014265LGN00_B1.TIF
- LC82220052014265LGN00_B2.TIF
- LC82220052014265LGN00_B3.TIF
- LC82220052014265LGN00_B4.TIF
- LC82220052014265LGN00_B5.TIF
- LC82220052014265LGN00_B6.TIF
- LC82220052014265LGN00_B7.TIF
- LC82220052014265LGN00_B8.TIF
- LC82220052014265LGN00_B9.TIF
- LC82220052014265LGN00_B10.TIF
- LC82220052014265LGN00_B11.TIF

Los anteriores archivos corresponden a cada una de las bandas de la imagen guardados en archivos de extensión 'geoTIFF', que es un estándar de metadatos de dominio público que permite que información georreferenciada sea encajada en un archivo de imagen de formato TIFF. Incluye información adicional sobre el tipo de proyección, sistema de coordenadas y todo lo necesario para que la imagen pueda ser automáticamente posicionada en un sistema de referencia espacial.

Durante la generación del producto, adicionalmente a las bandas se crea un archivo de metadatos L1G, que contiene información específica del producto solicitado. El archivo de metadatos (MTL) es un fichero de texto en formato '*Object Description Language*' (ODL). Los

archivos MTL contienen las constantes térmicas necesarias para convertir los datos TIRS de brillo a temperatura o de número digital a reflectancias. En nuestro caso, el archivo de metadatos es:

- LC82220052014265LGN00_MTL.txt

2.2 BANDAS

Como hemos comentado anteriormente, Landsat 8 dispone de 11 bandas multiespectrales. Incluye dos nuevas bandas, *Coastal/Aerosol* y *Cirrus*, que no estaban incluidas en ninguno de los anteriores Landsat. La banda térmica ha sido dividida en 2 bandas. Además todas las bandas han visto reducido su rango comparado con Landsat 7.

A continuación, tenemos la tabla correspondiente a las bandas de Landsat 8 con sus respectivos anchos de banda y resolución espacial con la que se obtienen:

Banda	Ancho (μm)	Resolución (m)
1 Coastal/Aerosol	0.433–0.453	30
2 Blue	0.450–0.515	30
3 Green	0.525–0.600	30
4 Red	0.630–0.680	30
5 NIR	0.845–0.885	30
6 SWIR 1	1.560–1.660	60
7 SWIR 2	2.100–2.300	30
8 Panchromatic	0.500–0.680	15
9 Cirrus	1.360–1.390	30
10 TIR 1	10.6–11.2	100*
11 TIR 2	11.5–12.5	100*

Tabla 5: Bandas de Landsat 8.

*Se obtienen con una resolución espacial de 100 metros, pero se remuestrean a 30 metros.

Una vez hemos clarificado todo lo necesario sobre Landsat 8, analizaremos sus bandas en profundidad poniendo un especial énfasis en la razón de existencia de la banda y en sus aplicaciones.

2.2.1 Banda 1 (0.433-0.453 μm ; Coastal/Aerosol)

La banda 1 detecta los azules y violetas intensos.

La luz azul es difícil de captar desde el espacio por que se dispersa fácilmente en pequeños trozos de polvo y agua en el aire, e incluso en moléculas de aire. Por eso, el cielo es azul y las montañas a lo lejos las vemos azules. De modo que, esta parte del espectro es difícil de captar con la suficiente sensibilidad para ser útil.

Para solventar esta dificultad, se ha incorporado esta nueva banda que ve el suelo a través de todos estos inconvenientes. De esta forma podemos observar la calidad del agua en lagos de poca profundidad y en zonas costeras así como detectar finas partículas de polvo y humo como concentraciones atmosféricas de aerosol.

2.2.2 Banda 2 (0.450–0.515 μm ; Blue)

La banda 2 corresponde al Azul del espectro visible.

Esta banda permite el mapeo de cuerpos de agua al atravesarlos y diferencia la superficie del suelo y de la roca de la vegetación así como diferencia entre vegetación perenne y caduca. También permite detectar zonas de carreteras o similares.

Tiene el inconveniente de ser susceptible a la dispersión de la atmósfera, convirtiéndola en la banda 'más ruidosa'.

2.2.3 Banda 3 (0.525–0.600 μm ; Green)

La banda 3 corresponde al Verde del espectro visible.

Está diseñada para evaluar el vigor de la vegetación sana. Al cubrir el pico de reflectancia verde, permite separar distintos tipos de vegetación. Además, esta banda puede diferenciar tipos de suelos: las tierras áridas, áreas urbanas y carreteras aparecen más brillantes y las zonas de bosques, vegetación y cultivos aparecen más oscuras. De igual forma, mide la calidad del agua al ser sensible a la turbidez del agua.

2.2.4 Banda 4 (0.433-0.453 μm ; Red)

La banda 4 corresponde al Rojo del espectro visible.

Esta banda es sensible a regiones de fuerte absorción de clorofila y posee una alta reflectancia para la mayoría de los suelos. Resulta muy útil para la clasificación de la cubierta vegetal, agricultura y uso del suelo pero no permite diferenciar agua de bosques, ya que ambos aparecen en tono oscuro.

2.2.5 Banda 5 (0.845–0.885 μm ; NIR)

La banda 5 mide el infrarrojo cercano o NIR.

Esta banda es especialmente importante para la ecología porque las plantas sanas lo reflejan; esto es, el agua de sus hojas refleja las longitudes de onda. Comparando con otras bandas, obtenemos índices de vegetación (NDVI) con mayor precisión que si sólo miráramos la vegetación visible. Así, esta banda del infrarrojo cercano sirve para distinguir tipos de vegetación.

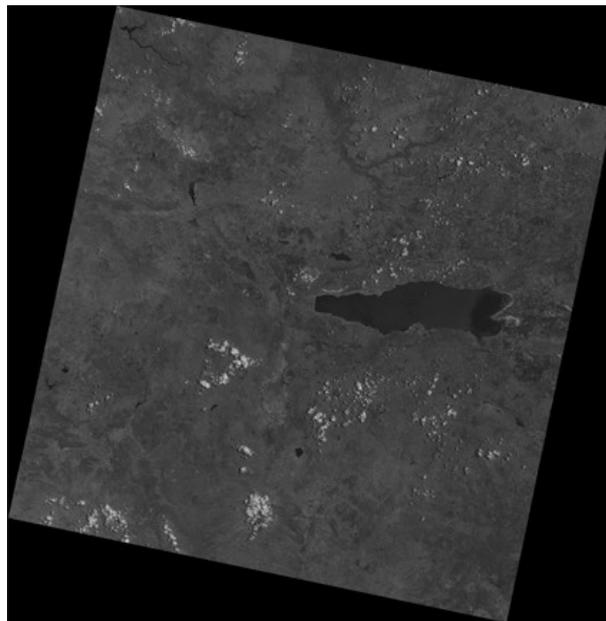


Figura 5: Banda NIR.

También nos permite diferenciar suelos húmedos de secos. Las zonas de cultivo y pastizales muestran una mayor reflectancia (tono más claro) que las zonas boscosas; mientras que las tierras áridas, zonas urbanas y carreteras se ven en tono oscuro. De esta forma, podemos determinar el contenido de la biomasa y elaborar mapas geológicos y geomorfológicos.

Esta banda es muy útil para la delimitación de zonas de agua, ya que el agua se absorbe en el infrarrojo cercano y realza el contraste tierra-agua.

2.2.6 Banda 6 (1.560–1.660 μm ; SWIR 1)

La banda 6 corresponde a una parte del infrarrojo de onda corta o SWIR.

Es sensible a la turgencia o cantidad de agua en las plantas lo cual nos permite diferenciar el suelo húmedo del suelo seco. Esta banda separa tierras forestales, tierras de cultivo y zonas de

agua. Las zonas boscosas aparecen más oscuras que las zonas de cultivo. Los cuerpos de agua también aparecen en un tono oscuro.

No obstante, esta banda no sirve para separar las áreas urbanas y las tierras de cultivo pero sí diferencia entre nubes y nieve.

2.2.7 Banda 7 (2.100–2.300 μm ; SWIR 2)

La banda 7 corresponde a otra parte del infrarrojo de onda corta o SWIR.

Su especialidad es la separación de tierra y agua. En cuanto a la longitud de onda, el agua tiene una fuerte absorción mientras que el suelo y la roca tienen una gran reflectancia. Las zonas urbanas, tierras de cultivo, carreteras, terrenos baldíos aparecen en un tono brillante. Los cuerpos de agua y los bosques en un tono oscuro.

Además, es especialmente útil para la geología. Rocas y suelos que se ven similares en otras bandas, tienen gran contraste en esta banda, lo cual posibilita realizar un 'mapeo hidrotermal' y la identificación de minerales.

2.2.8 Banda 8 (0.500–0.680 μm ; Panchromatic)

La banda 8 es la banda pancromática.

Su característica principal es tener una resolución espacial mayor que el resto de las bandas, una resolución de 15 metros. En vez de recoger los colores visibles por separado, combina estos en un solo canal, como las películas en blanco y negro, dado que su sensor puede ver más de una luz a la vez.

Su principal uso es combinarse con otras bandas para obtener imágenes de mayor calidad, y así permitir trabajar con imágenes de 15 metros de resolución en vez de 30.

2.2.9 Banda 9 (1.360–1.390 μm ; Cirrus)

La banda 9 se encarga de detectar nubes de tipo cirro.

Pocos instrumentos del espacio recogen esta parte del espectro que capta esta banda, porque la atmósfera absorbe casi todo ello. Landsat 8 ha convertido este inconveniente en algo ventajoso. Precisamente la tierra es apenas visible en esta banda, y cualquier cosa que aparezca debe reflejarse con gran intensidad o estar por encima de la mayor parte de la atmósfera.

De esta forma, esta banda es capaz de detectar nubes de tipo cirro. Este tipo de nubes son un gran inconveniente para las imágenes satelitales, porque sus bordes son difícilmente delimitables, haciendo que las imágenes tomadas a través de ellas puedan contener artefactos sin ninguna explicación obvia.

2.2.10 Bandas 10 (10.6-11.2 μ m; TIR 1) y 11 (11.5-12.5 μ m; TIR 2)

La bandas 10 y 11 son las bandas termales pertenecientes al infrarrojo termal o TIR.

En lugar de medir la temperatura del aire, como lo hacen las estaciones meteorológicas, miden la temperatura de la superficie que, generalmente, es más alta.

Así permiten determinar el estrés térmico de la vegetación, corrientes marinas o propiedades termales del suelo.

2.2.11 Banda BQA

La banda BQA es la banda del control de la calidad.

Es una banda artificial generada por NASA con datos adicionales donde cada píxel de la banda de control de calidad contiene un valor decimal que representa las combinaciones de bits de relleno de la superficie, la atmósfera, y las condiciones de los sensores que pueden afectar a la utilidad general de un píxel dado.

2.3 COMBINACIONES DE BANDAS

Al combinar distintas bandas de una imagen satelital multiespectral para obtener una imagen RGB, obtenemos distintos resultados que pueden ayudarnos a identificar mejor los objetivos que tratamos de estudiar, facilitándonos la clasificación de la imagen, mostrándonos resultados visualmente mejores, eliminando ruido,...

2.3.1 Combinaciones de tres bandas

Estas combinaciones de bandas se obtienen de la unión de tres bandas como RGB; es decir, una banda como 'Red', otra como 'Blue' y otra como 'Green'. Con distintas bandas para cada componente de la combinación obtenemos distintos resultados, como el color natural o el color

infrarrojo. En la siguiente tabla se muestran las diez combinaciones más comunes para Landsat 8:

Nombre	Bandas
Color natural	4 3 2
Falso color (urbano)	7 6 4
Color infrarrojo (vegetación)	5 4 3
Agricultura	6 5 2
Penetración atmosférica	7 6 5
Vegetación saludable	5 6 2
Tierra/agua	5 6 4
Natural con eliminación atmosférica	7 5 3
Infrarrojo de onda corta	7 5 4
Análisis de vegetación	6 5 4

Tabla 6: Combinaciones comunes de tres bandas en Landsat 8.

Un ejemplo de combinación RGB es la siguiente:



Figura 6: Combinación RGB en falso color.

Estas combinaciones nos ofrecen buenos resultados visuales, ayudando a identificar los elementos del mapa gracias a los colores obtenidos. Pero este tipo de combinaciones no son adecuadas para clasificación, por lo tanto sólo serán empleadas en la visualización.

2.3.2 Operaciones entre bandas

Estas combinaciones de bandas se obtienen de la realización de operaciones matemáticas sencillas empleando dos bandas. Hay dos tipos de operaciones:

- Banda A / Banda B
- (Banda A - Banda B) / (Banda A + banda B)

A continuación, se enumeran las operaciones entre bandas más importantes:

Nombre	Operación
R/NIR	Banda 4 / Banda 5
SWIR1/SWIR2	Banda 6 / Banda 7
NIR/SWIR1	Banda 5 / Banda 6
SWIR1/NIR	Banda 6 / Banda 5
NDVI	$(\text{Banda 5} - \text{Banda 4}) / (\text{Banda 5} + \text{banda 4})$
NDWI1*	$(\text{Banda 5} - \text{Banda 6}) / (\text{Banda 5} + \text{banda 6})$
NDWI2	$(\text{Banda 3} - \text{Banda 5}) / (\text{Banda 3} + \text{banda 5})$
NDWI3	$(\text{Banda 4} - \text{Banda 6}) / (\text{Banda 4} + \text{banda 6})$
NDWI4	$(\text{Banda 3} - \text{Banda 6}) / (\text{Banda 3} + \text{banda 6})$

Tabla 7: Operaciones más importantes entre dos bandas en Landsat 8.

Estas combinaciones de bandas son muy utilizadas en el procesamiento de imágenes satelitales multiespectrales al estar ligadas directamente con ciertas propiedades. Algunas de estas propiedades (las directamente relacionadas con la presencia o detección de cuerpos de agua) son las siguientes:

- **R/NIR:** Distingue vegetación, agua y cultivos de regadío. Enfatiza bosques y tierras baldías. La vegetación y los bosques exhiben una alta reflectancia en la región del infrarrojo cercano (0.76 - 0.90 μm) y una fuerte absorción en la zona del rojo (0.63 - 0.69 μm).
- **SWIR1/SWIR2:** Separa tierra y agua. El suelo exhibe una alta absorción en la banda SWIR2 (2.08 - 2.35 μm) y una alta reflectancia en la banda SWIR1 (1.55 - 1.75 μm). La tierra aparece con un tono claro y el agua con un tono oscuro. También enfatiza la

presencia de humedad en los cultivos de regadío, porque las bandas SWIR son sensibles a la variación de la humedad en suelos y vegetación.

- **NIR/SWIR1:** Enfatiza los cuerpos de agua, vegetación y la presencia de humedad en las zonas de regadío. El agua tiene una alta absorción en la región del infrarrojo cercano NIR y una alta reflectancia en la SWIR1.
- **SWIR1/NIR:** Es una alternativa a su operación inversa. Se suelen presentar ambos cocientes.
- **NDVI:** NDVI (*'Normalized Difference Vegetation Index'*) o índice de vegetación de diferencia normalizada. Es un índice que tiene en cuenta el rojo visible y la banda del infrarrojo cercano. Mide la vegetación, su condición y densidad: biomasa, LAI (*'leaf area index'*) y porcentaje de cubierta vegetal.

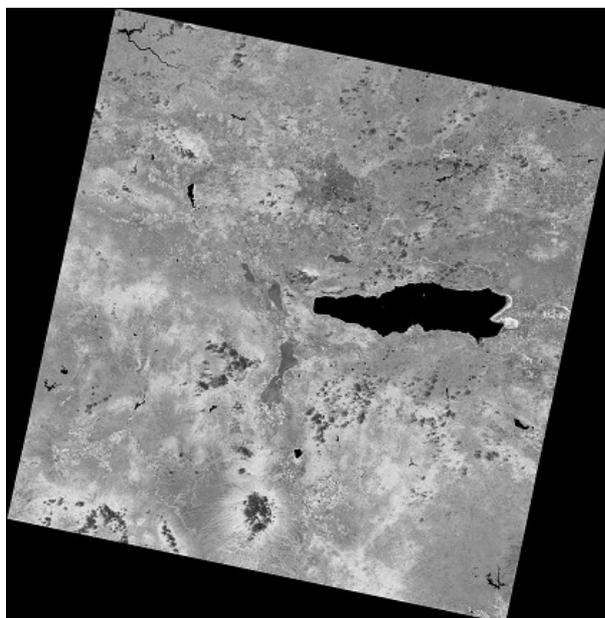


Figura 7: Índice NDVI.

Los índices NDWI ('Normalized Difference Water Index'*) miden la humedad del suelo. Al igual que NDVI, su rango es [-1,1]. Existen varias versiones para estos índices y, a continuación, presentaremos las cuatro más utilizadas:

- **NDWI1:** La banda SWIR1 se encuentra en una región donde el agua tiene una alta absorción mientras que la banda NIR, en esta región, no absorbe el agua. Por ello, este índice tiene en cuenta el contenido de agua de la vegetación. Fue presentado por Gao [P4] que en un principio empleaba este índice en detección de vegetación. Además, inicialmente empleaba dos bandas NIR, con un ancho de onda alrededor de $0.86 \mu\text{m}$ la

primera y de 1.24 μm la segunda. La actual banda SWIR1 ha sustituido a esta segunda banda NIR.

- **NDWI2:** La utilización de la banda 'Green' maximiza la reflectancia del agua. Por otro lado, se minimiza la poca reflectancia de la banda NIR para cuerpos de agua. Este índice aprovecha la gran reflectancia de la banda NIR en la vegetación terrestre y en cuerpos de tierra. De esta forma, los valores mayores a cero son agua y, en caso contrario, tierra. Este índice fue propuesto por McFeeters [P5].
- **NDWI3:** Es uno de los índices empleados por Rogers and Kearney [P6]. Este índice tiene en cuenta la banda 'Red' que es la única que refleja el agua mejor que la banda SWIR1. Esta combinación da valores entre [-1, 1] siendo los valores altos agua, los intermedios vegetación y los bajos tierra.
- **NDWI4:** Xu [P9] demostró cómo el índice de McFeeters [P5] era incapaz de separar completamente cuerpos de agua con zonas urbanas ya que ambas mostraban valores positivos porque la reflectancia de la banda NIR es menor que la reflectancia de la banda 'Green'. Para solventar estas deficiencias, Xu [P9] reemplazó la banda del NIR por la SWIR1.

2.4 TÉCNICAS DE DETECCIÓN DE CUERPOS DE AGUA

A la hora de detectar cuerpos de agua, existen un gran número de técnicas. A continuación, explicaremos las más importantes:

2.4.1 Density slicing

Esta técnica emplea una única banda en niveles digitales y se encuentra representada en escala de grises, en un rango [0, 255]. La técnica consiste en dividir este rango en diferentes trozos (no teniendo porqué ser del mismo tamaño) y asignar a cada intervalo un color arbitrario. De este modo, la banda en escala de grises pasa a ser una imagen en color con cierta cantidad de colores, facilitando el posterior análisis.

2.4.2 Umbral sobre una única banda

Se escoge una banda y se binariza utilizando cierto valor umbral. La dificultad de esta técnica reside en encontrar el valor umbral ideal. Existen distintas técnicas para obtenerlo: empleando el histograma, algoritmos matemáticos,...

Las bandas escogidas para esta técnica suelen pertenecer al intervalo cercano (NIR) o intervalo medio (SWIR) para la detección de cuerpos de agua.

2.4.3 Umbral sobre varias bandas

Este método es similar al anterior. Se eligen dos o más bandas y se suman. Posteriormente, se busca el umbral ideal para dicha suma de bandas, generalmente, mediante el histograma. Este método es uno de los métodos empleados por Zahidul Islam [P8] en la detección de inundaciones.

Las bandas a emplear suelen ser las mismas que en el caso anterior; esto es, bandas del NIR y del SWIR.

2.4.4 Umbral sobre combinaciones de bandas

En este caso, se utiliza la misma técnica vista anteriormente pero, en este caso, aplicada a ciertas combinaciones de bandas. Estas combinaciones son las vistas en el apartado 2.3.2. Dada una combinación de bandas, se umbraliza y se asignan unos valores como agua y otros como tierra.

En el caso de los índices de agua, generalmente el umbral se encuentra en el valor 0, como comentan varios autores, entre ellos McFeeters [P5]. Otros, en cambio, proponen el valor 0 u otro, pero ajustándolo según las características del entorno; es decir, un umbral dinámico. Esto mismo proponen Lei Ji, Li Zhang, and Bruce Wylie [P7].

2.4.5 Algoritmos de clasificación

Empleando distintos algoritmos de clasificación se realiza una clasificación supervisada o no supervisada para obtener una imagen binaria, donde un valor indique agua y el otro tierra.

Este método puede aplicarse empleando una banda, una combinación de bandas o varias bandas y/o combinaciones. Según lo empleado pueden aplicarse distintas técnicas de entrenamiento, distintos algoritmos,...

2.5 SOFTWARE A UTILIZAR

A lo largo del proyecto se han utilizado distintos *software* o programas informáticos. Entre ellos destacamos los siguientes:

- GRASS GIS
- Quantum GIS
- Weka
- R

Para facilitar su lectura y entendimiento posteriormente, se introducirán brevemente los cuatro *software* mencionados y sus principales usos.

2.5.1 GRASS GIS

GRASS GIS, comúnmente llamado GRASS ('*Geographic Resources Analysis Support System*') es un Sistema de Información Geográfica (GIS) gratuito y de código abierto, usado para el manejo y análisis de información geoespacial, procesamiento de imagen, producción de gráficos y mapas, modelado espacial, y visualización.

En nuestro proyecto emplearemos GRASS en la realización de diversas tareas sobre las imágenes satelitales multiespectrales, como visualización, corrección radiométrica, recorte o extracción de información.

Este en las últimas versiones ha mejorado sensiblemente la experiencia de los usuarios incluyendo interfaces gráficas más amigables y fáciles de usar. A la hora de trabajar con GRASS observaremos como dispone de una pantalla de inicio donde se define la localización del proyecto y sus directorios de mapas.

Dentro de GRASS, tras la pantalla de inicio, apreciamos dos ventanas. La primera, llamada '*Layer Manager*', es la ventana de trabajo. En esta se muestran las capas, se realizan los procesos, se encuentran accesibles todas las herramientas,... La segunda, llamada '*Map Display*', es la ventana de visualización donde se muestran las imágenes, los mapas, los gráficos,... De este modo en el '*Layer Manager*' se realizan los procesos y en '*Map Display*' se muestran los resultados.

La ventana '*Layer Manager*' dispone de cuatro secciones: '*Map Layer*' donde tenemos los elementos de trabajo, es decir, las capas; '*Command Console*' donde podremos ejecutar comandos de GRASS; '*Search modules*' donde se encuentran jerárquicamente divididas todas las herramientas de GRASS; y '*Python shell*', donde se puede introducir y ejecutar código '*Python*'.

También disponemos de menús donde están almacenadas todas las herramientas o funcionalidades además de iconos de acceso directo a las más importantes.

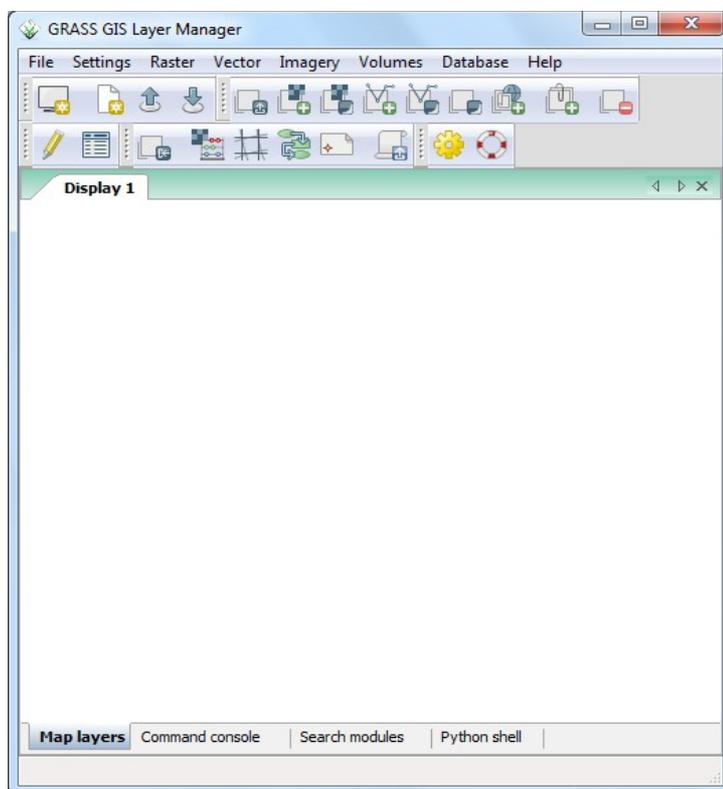


Figura 8: Ventana principal de GRASS.

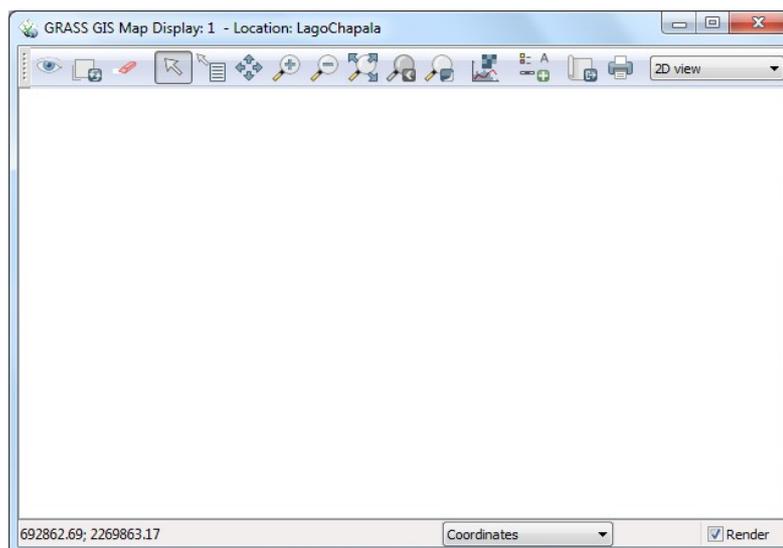


Figura 9: Ventana de visualización de GRASS.

En GRASS para ejecutar una funcionalidad generalmente tendremos dos opciones: ejecutar la función mediante la interfaz gráfica o introducir el comando correspondiente a esa función. GRASS en todo momento nos proporciona el comando equivalente a cualquier acción realizada mediante la interfaz gráfica. De esta forma es muy fácil mostrar cómo se realiza una operación en GRASS junto a su comando equivalente.

Los comandos de GRASS tienen la forma 'i.landsat.toar'. Una o dos letras de inicio seguidas de un punto, seguido de un nombre y opcionalmente seguido de otro punto y otro nombre. Los comandos los mostraremos entre llaves como aparecen en los menús de GRASS.

GRASS se encuentra en inglés, pero algunas opciones se encuentran en castellano. Esto puede conllevar confusión. Trataremos de mostrar todo completamente en inglés para evitar la confusión, pero puede que aparezcan algunos elementos en castellano.

GRASS principalmente será empleado en el preprocesado. Para realizar todos los procesos necesarios de adecuar la imagen satelital multispectral inicial, a las necesidades de las tareas posteriores como la clasificación.

2.5.2 Quantum GIS

Quantum GIS o QGIS es un Sistema de Información Geográfica (GIS) de *software* libre y de código abierto que permite crear, visualizar, analizar y publicar información geoespacial. Soporta numerosos formatos y funcionalidades de datos vector, datos *ráster* y bases de datos.

La principal tarea a realizar en QGIS será la obtención las regiones de interés (ROI). Con las cuales trabajaremos durante la clasificación. Estas regiones las extraeremos mayormente mediante el *plugin* de QGIS '*Semi-Automatic Classification*'. Este nos facilitará la obtención de zonas similares al seleccionar píxeles con valores espectrales similares en un conjunto de bandas.

GRASS y QGIS son dos *software* totalmente compatibles. Disponen de multitud de posibilidades para trabajar con ambos de forma simultánea. QGIS se encuentra totalmente en castellano. Exceptuando los *plugin*.

2.5.3 R (lenguaje de programación)

R es un lenguaje y un entorno de programación para análisis estadístico y gráfico. R es un lenguaje interpretado, los comandos escritos mediante teclado son ejecutados directamente sin necesidad de construir ejecutables. Se trata de un *software* libre, resultado de la implementación GNU ('*GNU's Not Unix*') del lenguaje S.

R es uno de los lenguajes más utilizados en estadística. Además abarca otros muchos campos, principalmente debido a la gran cantidad de librerías disponibles, entre las cuales se encuentran librerías de tratamiento de imágenes satelitales, clasificación supervisada,...

R será la principal herramienta a utilizar durante la clasificación supervisada y no supervisada (si se diera el caso). Además será un 'enlace' en el proyecto. Es decir, R se encargará de unir todo el proceso realizado a lo largo del proyecto y de obtener resultado final.

2.5.4 Weka

Weka ('*Waikato Environment for Knowledge Analysis*') es un *software* empleado en aprendizaje automático y minería de datos escrito en '*Java*' y desarrollado en la Universidad de Waikato. Weka es un software libre distribuido bajo licencia GNU-GPL.

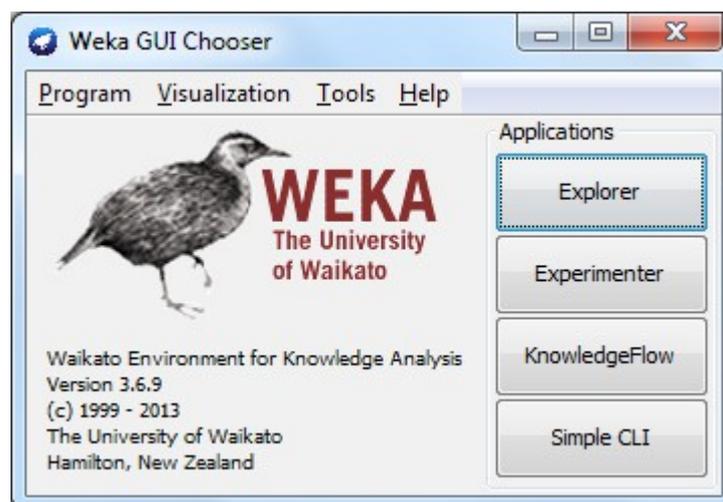


Figura 10: Pantalla de inicio de Weka.

Weka nos permite analizar bases de datos así como sus conjuntos de entrenamiento de una forma fácil y sencilla. Podría haber sido una herramienta a considerar a usar como principal, pero dados los datos a procesar, Weka tendría imposible manejar tal gran cantidad de datos. Aun así, nos servirá para realizar ciertos análisis o pequeñas pruebas de forma sencilla sin tener que escribir código en lenguaje R.

Hemos de remarcar esta herramienta, debido a que emplearemos varios de sus clasificadores en el lenguaje R mediante ciertos paquetes de éste.

2.6 ALGORITMOS DE CLASIFICACIÓN

A continuación presentaremos los algoritmos de clasificación elegidos a emplear en el proyecto. Estos serán tres: J48, NB y KNN.

2.6.1 J48

J48 es una implementación *open source* en el lenguaje de programación 'Java' del algoritmo C4.5, para la herramienta Weka de minería de datos.

El algoritmo C4.5 fue desarrollado por JR Quinlan en 1993, como una extensión (mejora) del algoritmo ID3 que desarrollo en 1986. Este algoritmo genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero ('*depth-first*').

El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que obtiene la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con '*n*' resultados, siendo '*n*' el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. En cada nodo, el sistema debe decidir cuál prueba escoge para dividir los datos.

2.6.2 NB (Naive Bayes)

El clasificador Naive Bayes es un clasificador probabilístico fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales. Es a causa de estas simplificaciones, que se suelen resumir en la hipótesis de independencia entre las variables predictoras, que recibe el apelativo de ingenuo.

En simples términos, un clasificador Naive Bayes asume que la presencia o ausencia de una característica particular, no está relacionada con la presencia o ausencia de cualquier otra característica, teniendo en cuenta la variable de clase. Por ejemplo, una fruta puede ser considerada una manzana si es roja, redonda y alrededor de 3 cm de diámetro. El clasificador Naive Bayes considera cada una de esas funciones para contribuir de manera independiente a la probabilidad de que esta fruta sea una manzana, independiente de la presencia o ausencia de las otras características.

Para algunos tipos de modelos de probabilidad, el clasificador Naive Bayes puede ser entrenado de manera muy eficiente en un entorno de entrenamiento supervisado. En muchas aplicaciones prácticas la estimación de parámetros para los modelos Naive Bayes utiliza el

método de máxima verosimilitud, es decir, se puede trabajar con el modelo Naive Bayes sin aceptar la probabilidad bayesiana o el uso de cualquiera de los métodos bayesianos.

Una ventaja del clasificador de Bayes ingenuo es que sólo se requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros (las medias y las varianzas de las variables) necesarias para la clasificación. Como se asume que las variables son independientes, sólo es necesario determinar las varianzas de las variables de cada clase y no toda la matriz de covarianza.

2.6.3 KNN

KNN ('*K nearest neighbours*') es un método no paramétrico usado en clasificación y regresión. Estima la clase de un individuo en función de los '*k*' vecinos más cercanos mediante un sistema de votación, donde la mayoría determina la clase.

Las reglas de clasificación por vecindad están basadas en la búsqueda en un conjunto de prototipos de los '*k*' prototipos más cercanos al patrón a clasificar. Se basa en la suposición de que los prototipos más cercanos tienen una probabilidad a posteriori similar. Las predicciones se realizan basándose en los ejemplos más parecidos al que hay que predecir.

Para medir la proximidad es necesaria una métrica. Por razones computacionales se suele emplear la distancia Euclídea. Aunque existen multitud de posibilidades, como la distancia de Manhattan, la Euclídea ponderada,...

El entrenamiento de un clasificador KNN en un sistema supervisado consiste en la búsqueda del '*k*' óptimo dado el conjunto de entrenamiento.

'3'

Capítulo 3: Diseño

3.1 INICIO

El sistema operativo a emplear en el proyecto será *Windows*. No se descarta la posibilidad de emplear otros sistemas operativos a lo largo del proyecto, pero inicialmente y principalmente se empleará *Windows* y más concretamente, '*Windows 7 Professional 64 bits*'.

3.1.1 Primeros pasos

Inicialmente disponemos de un archivo tipo '*.tar.gz*', '*LC80290462013161LGN01.tar.gz*', el cual contiene la imagen satelital multiespectral. De este obtenemos los siguientes archivos:

- *LC80290462013161LGN01_B1.TIF*
- *LC80290462013161LGN01_B2.TIF*
- *LC80290462013161LGN01_B3.TIF*
- *LC80290462013161LGN01_B4.TIF*

- LC80290462013161LGN01_B5.TIF
- LC80290462013161LGN01_B6.TIF
- LC80290462013161LGN01_B7.TIF
- LC80290462013161LGN01_B8.TIF
- LC80290462013161LGN01_B9.TIF
- LC80290462013161LGN01_B10.TIF
- LC80290462013161LGN01_B11.TIF
- LC80290462013161LGN01_BQA.TIF
- LC80290462013161LGN01_MTL.txt

En total disponemos de 11 bandas espectrales, la banda del control de calidad y el archivo de metadatos.

3.1.2 Software a emplear

A continuación instalamos el *software* necesario para empezar a trabajar. Empezando con los Sistemas de Información Geográfica (GIS), GRASS GIS y Quantum GIS, concretamente las siguientes versiones:

- GRASS GIS 7.0 (32 bit) para *Windows*.
- GRASS GIS 6.4.3 (32 bit) para *Windows*.
- Quantum GIS 2.0.1 (32 bit) para *Windows*.

También instalaremos los *software* a emplear en clasificación, R y Weka:

- R 3.0.2 (64 bits) para *Windows*.
- Rstudio 0.97 (64 bits) para *Windows*.
- Weka 3.6.9 (32 bits) para *Windows*.

Una vez este el *software* instalado correctamente empezaremos trabajando con GRASS en los primeros pasos a realizar. Posteriormente utilizaremos QGIS y después R y Weka.

3.2 LECTURA DE DATOS INICIALES

Empezaremos leyendo los datos iniciales, es decir, introduciendo las bandas de nuestra imagen. Esto lo haremos empleando GRASS GIS, para lo cual primero de todo tendremos que crear una nueva localización en GRASS. En la pantalla de inicio presionamos en '*Location Wizard*'.

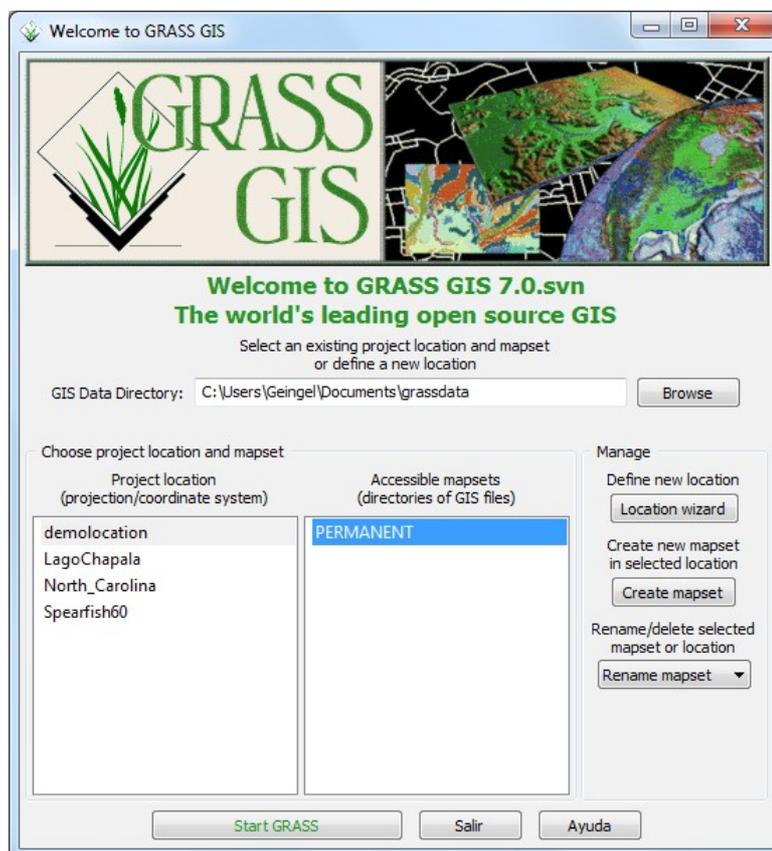


Figura 11: Pantalla de inicio de GRASS.

- El primer paso será elegir la nueva localización donde escribimos el nombre 'LagoChapala'.
- Elegimos el método de creación '*Read projection and datum terms from a georeferenced data file*', mediante el cual se creará el directorio con la información de una de las bandas.
- Seleccionamos la banda 1, que proporcionará a GRASS toda la información sobre el tipo de proyección, sistema de coordenadas,...
- Tras crearse aceptamos los siguientes pasos. En el último paso nos preguntará si queremos crear un *mapset* (directorio de mapas), indicamos que no. De este modo usaremos el *mapset* por defecto, llamado 'PERMANENT'.

Tras completar estos pasos elegimos nuestro proyecto, 'LagoChapala', y el *mapset* 'PERMANENT' e iniciamos el programa pulsando 'Start GRASS'.

El siguiente paso será la carga de la imagen y todas sus bandas. Empezaremos creando nuevo proyecto 'File->Workspace->New' o 'Ctrl + N'. Seguido hemos de importar las bandas, 'File -> Import raster data -> Common formats import' o [r.in.gdal].

- Seleccionamos 'Directory' como 'Source Type'.
- Automáticamente aparecerá 'GeoTIFF' como formato por defecto.
- Seleccionamos todas las bandas del directorio donde estén almacenadas.
- Marcamos la opción 'Extend region extends based on new dataset'.
- Opcionalmente marcamos 'Keep band numbers instead of using band color names'.
- Todo lo demás lo dejamos como está y presionamos 'Import'.

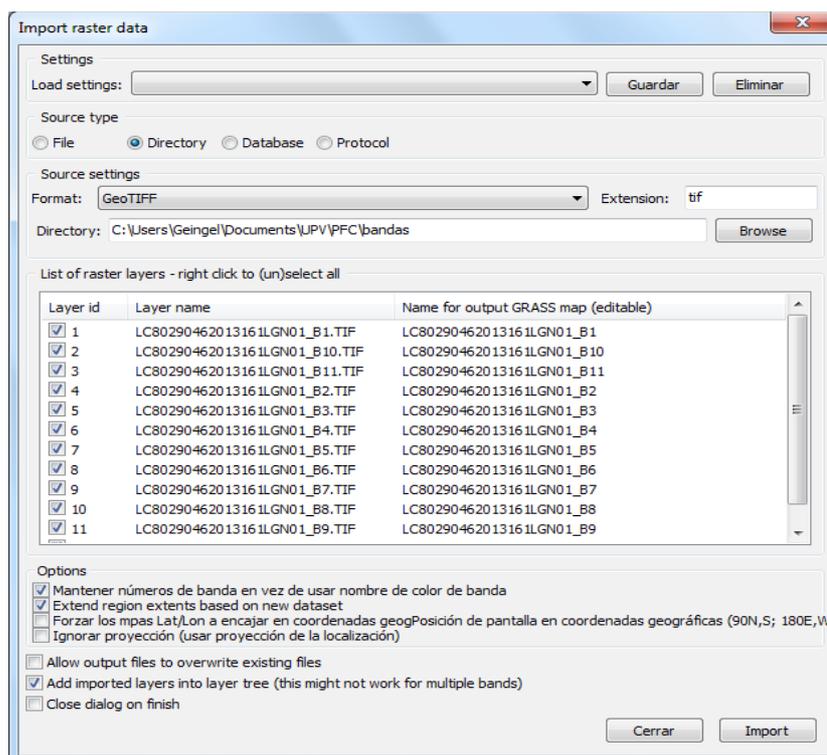


Figura 12: Importar bandas en GRASS.

Una vez hecho esto se habrán importado las bandas en GRASS. Si anteriormente seleccionamos la opción 'Add imported layers into layer tree', se habrán cargado en el proyecto actual. En caso contrario podemos cargarlas de la siguiente forma:

- 'File -> Map display -> Add multiples raster or vectors' o 'Ctrl + Shift + L'.

- Como tipo elegimos 'Raster'.
- Seleccionamos nuestro *mapset*, 'PERMANENT'.
- Marcamos las bandas a cargar y aceptamos.

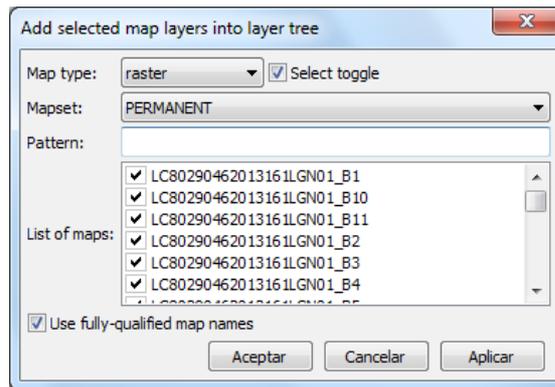


Figura 13: Carga de bandas en GRASS.

Así ya tenemos nuestras bandas listas para trabajar en GRASS. En el siguiente paso comenzaremos a trabajar con ellas.

3.3 TRATAR VALORES NULOS

La primera operación a realizar en GRASS será el tratamiento de valores nulos. Como bien sabemos la imagen satelital es 'un rombo dentro de un cuadrado'. Todos los valores externos al 'rombo' son nulos. Para evitar posibles problemas a posteriori debidos a los valores nulos hemos de tratarlos. El tratamiento consistirá en asignar el valor -9999 a los valores nulos.

Se ha elegido el valor -9999 debido a que los índices de vegetación y de agua (que calcularemos posteriormente) se encuentran en el rango de valores [-1, 1]. En esos casos si el valor nulo fuera 0 podría causarnos problemas.

El tratamiento de valores nulos en GRASS la podemos realizar mediante el comando [r.null]. Seguiremos los siguientes pasos:

- 'Raster -> Develop raster map -> Manage NULL values'
- Elegimos la banda a procesar.
- En 'Modify' asignaremos el valor -9999 como valor a sustituir.
- Ejecutamos la misma operación para todas las bandas.

Una vez finalizado podemos pasar a la siguiente fase.

3.4 TRANSFORMACIÓN A REFLECTANCIAS

Como bien sabemos, nuestra imagen satelital es un producto L1T, el producto estándar de Landsat 8. Éste consiste en una serie cuantificada, calibrada y escalada de niveles digitales (ND), los cuales representan los datos de una imagen multispectral. Los datos de las bandas están derivados en 16 bits en formato no cifrado, pueden ser reescalados a los valores de reflectancia TOA (*Top Of Atmosphere*) (bandas 1-9) o radiación (bandas 1-11) empleando los coeficientes radiométricos provistos en el archivo de metadatos '*MTL.txt*'.

Realizaremos la transformación a reflectancia en GRASS. Esta transformación es necesaria, si no la realizamos estaremos trabajando con imágenes inadecuadas obteniendo resultados no validos. Podemos realizar la transformación mediante el comando [i.landsat.toar].

A esta funcionalidad accedemos desde '*Imagery -> Satellite images tools -> Landsat DN to radiance/reflectancia*'. Realizamos los siguientes pasos:

- En el apartado '*Required*', en el primer campo introducimos el prefijo de nuestras bandas '*input*', el cual es 'LC80290462013161LGN01_B'. Sabiendo este prefijo GRASS realizará la transformación a todas las bandas de nuestra imagen.
- En el segundo campo '*output*' introducimos el prefijo que deseamos para las imágenes transformadas, este será 'LC80290462013161LGN01_refl'.

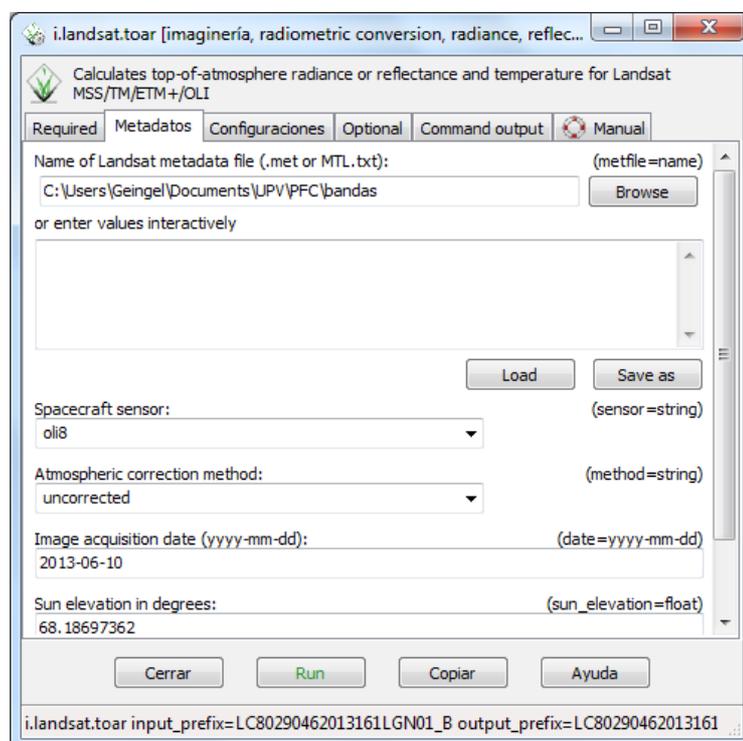


Figura 14: Propiedades conversión a reflectancias.

- En la pestaña 'Metadata', seleccionamos el archivo de metadatos. En este se encuentra toda la información necesaria para realizar la transformación. En nuestro caso se llama 'LC80290462013161LGN01_MTL.txt'.
- Elegimos el sensor de nuestro satélite '*Spacecraft sensor*', el que será 'oli8' (u ot8, dependiendo de la versión).
- '*Atmospheric correction method*' será '*uncorrected*' ya que las imágenes se encuentran sin correcciones de este tipo.
- Los demás datos que nos piden '*Image acquisition date*' y '*Sun elevation in degrees*' (omitiremos '*Image creation date*') se encuentran en el archivo de metadatos. GRASS suele tomar estos datos automáticamente, pero en algunas versiones no es así; por lo tanto buscaremos en el archivo de metadatos esta información y la introducimos.

GRASS nos proporciona el comando correspondiente a esta acción. El cual es:

```
i.landsat.toar
input_prefix=LC80290462013161LGN01_B
output_prefix=LC80290462013161LGN01_refl
metfile=C:\Users\Geingel\Desktop\PF\LC80290462013161LGN01_MTL.txt
sensor=oli8
date=2013-06-10
sun_elevation=68.18697362
```

Realizando los pasos mediante la interfaz gráfica o introduciendo el comando obtendríamos los mismos resultados. Al igual que todos los comandos primero comienza con el nombre de comando [i.landsat.toar] seguido de cada uno de los parámetros introducidos. Algunos parámetros son obligatorios y por defecto vienen incluidos en la función, otros en cambio son opcionales. Estos parámetros irán seguidos de un '=' y del dato proporcionado.

Ejecutamos el comando y obtenemos una nueva imagen satelital multiespectral con todas sus bandas corregidas. Posteriormente trabajaremos única y exclusivamente con la imagen corregida. GRASS ha creado las nuevas bandas en el directorio de mapas. Desde el cual podremos cargarlas mediante '*Add multiples raster or vectors*'.

3.5 GENERACIÓN DE ÍNDICES DE VEGETACIÓN Y AGUA

Crearemos un nuevo proyecto en el cual cargaremos las bandas de la imagen corregida. A continuación realizaremos ciertas operaciones entre las bandas. Para realizar estas operaciones utilizaremos la calculadora *ráster* ('*Raster Map Calculator*'). Podemos acceder a esta mediante '*Raster -> Raster map calculator*' o [r.mapcalc] (figura 15).

Realizaremos todas las operaciones vistas en el apartado 2.3.2. Obteniendo 9 nuevas bandas en el proceso. Para realizar estas operaciones podremos ayudarnos de la interfaz gráfica y realizar ecuaciones con la siguiente forma:

```
float(LC80290462013161LGN01_ref14@PERMANENT -
LC80290462013161LGN01_ref14@PERMANENT) /
float(LC80290462013161LGN01_ref15@PERMANENT +
LC80290462013161LGN01_ref14@PERMANENT)
```

El resultado de esta operación sería el NDVI. Dada la naturaleza de los datos es necesario incluir *float* en la ecuación para evitar posibles errores (como división entera) al dividir. Hemos de poner el nombre designado a cada combinación en el '*Output*'.

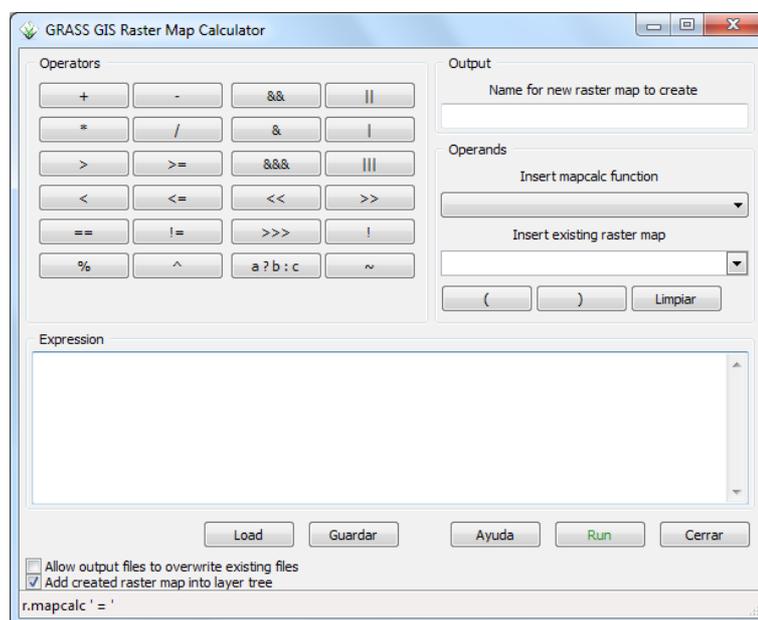


Figura 15: Calculadora *ráster* de GRASS.

Tras ejecutar cada operación en el directorio de GRASS se crearán los archivos necesarios para posteriormente poder cargar los índices creados como cualquier banda. A partir de ahora trabajaremos con los índices creados y las bandas corregidas conjuntamente.

Tras crear los índices, si observamos su resultado visual, obtenemos visualizaciones 'extrañas' en algunos casos. El '*Map Display*' de GRASS los muestra en color con colores muy saturados, lo cual nos puede inducir a pensar que hemos cometido algún error de cálculo. Podemos cambiar el rango de colores para obtener una visualización adecuada. Realizamos lo siguiente:

- Accedemos a '*Raster -> Manage colors -> Color tables*' o [*r.colors*].
- Pestaña '*Map*' elegimos una o varias bandas.
- Pestaña '*Define*' en '*Name of the color table*' elegimos un degradado de color.

Los colores asignados solo afectan a la visualización de GRASS. No alteran la información de los índices.

3.6 GENERACIÓN DE UNA SUB-IMAGEN

A continuación generaremos una sub-imagen a partir de la imagen satelital multiespectral, para posteriormente trabajar con una imagen más pequeña. Debido al tamaño descomunal de la imagen sus archivos son de gran tamaño y en consecuencia las operaciones sobre la imagen son muy costosas. La creación de una imagen más pequeña nos servirá para trabajar más cómodamente y realizar ciertas pruebas que posteriormente extrapolaremos a la imagen principal.

Para esto primero iremos a la ventana de visualización para visualizar el recorte deseado. Es decir, enfocar, mover,... hasta obtener la imagen que queremos tener tras el recorte. La extensión elegida será desde (647323.25, 743338.62) hasta (2218865.35, 2266086.02).

Una vez fijada la vista, pulsamos en un icono de la ventana con forma de lupa junto a un mapa. Este llamado '*Various zoom options*' tiene varias opciones, seleccionamos '*Set computational region from display extend*'.

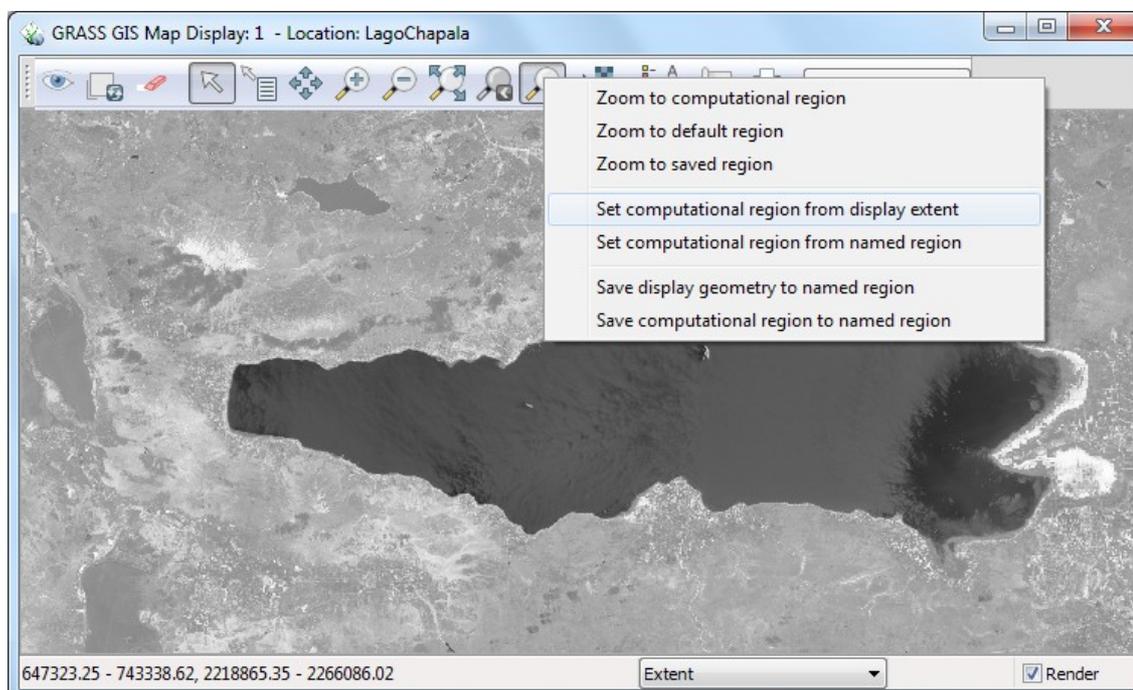


Figura 16: Recorte a aplicar a la imagen.

Sin modificar la vista nos dirigimos al menú a '*Raster -> Develop raster map -> Resample using nearest neighbor*' o mediante el comando [r.resample], donde introduciremos cada una de las bandas, para obtener su correspondiente recorte. Estos recortes serán guardados en el directorio de GRASS como las bandas. A los nombres de estos recortes les hemos añadido el sufijo '*_crop*' al nombre previo que tuvieran. Posteriormente en la pantalla de visualización, podemos observar como los recortes están perfectamente georeferenciados.

3.6.1 Exportar

Tras el recorte exportaremos todas las bandas de la imagen y sus correspondientes versiones recortadas, para lo cual accedemos a '*File -> Export raster map -> Common export formats*' o [r.out.gdal]. Exportaremos todo en formato 'GeoTIFF'. Este formato nos mantendrá toda la información de la imagen, como la posición georeferencial. Hemos de marcar la opción 'No escribir tabla de colores estándar GDAL', dado que no nos interesa que se escriba.

Así obtendremos la imagen definitiva y su sub-imagen, con todas sus bandas (20 exactamente), con las cuales trabajaremos posteriormente.

3.7 GENERACIÓN DE LOS ROI DESDE QUANTUM GIS

En nuestro siguiente paso utilizaremos el *software* Quantum GIS para la obtención de las regiones de interés (ROI). Para esta tarea emplearemos el *plugin* '*Semi-Automatic Classification 2.2.2*' de QGIS. Éste nos facilitará la creación de los ROI, de forma semi-automática.

Previamente hemos de instalar el *software* OSGeo4W, para el correcto funcionamiento del *plugin*. Seguidamente instalamos el *plugin* en QGIS. Tras esto tendremos que configurar las opciones de QGIS para que todo funcione correctamente.

Una vez esté todo listo empezaremos añadiendo las bandas recortadas a QGIS. En esta ocasión solo emplearemos las bandas de la 2 a la 7 (ambas incluidas), dado que son las bandas más adecuadas para la generación de ROI.

Podemos añadir las bandas mediante '*Capa->Añadir capa ráster...*'. Buscamos las bandas y las añadimos. También podemos arrastrar las capas desde su carpeta al programa al panel '*Capas*'.

Para los ROI escogeremos distintas zonas de interés, que serán de 5 tipos: agua, ciudad, cultivo, bosque y desierto. De cada tipo tomaremos 5 muestras, única y exclusivamente de ese tipo (no puede haber una muestra perteneciente a más de un tipo). Estas muestras han de ser lo más variadas posibles dentro de su tipo. Es decir, en el caso del agua, obtener muestras de lagos o zonas diferentes, para así poder abarcar la máxima variabilidad posible.

El siguiente paso será seleccionar las bandas a utilizar en el *plugin*. En el panel (del *plugin*) '*creation*' en la parte '*Input*', seleccionamos '*Band Set*'. Seleccionamos todas las bandas y las añadimos al '*Set*' con '*Add rasters to set*'. Seguidamente creamos un archivo '*shapefile*' ('*.shp*') mediante la opción en '*New shapefile*'.

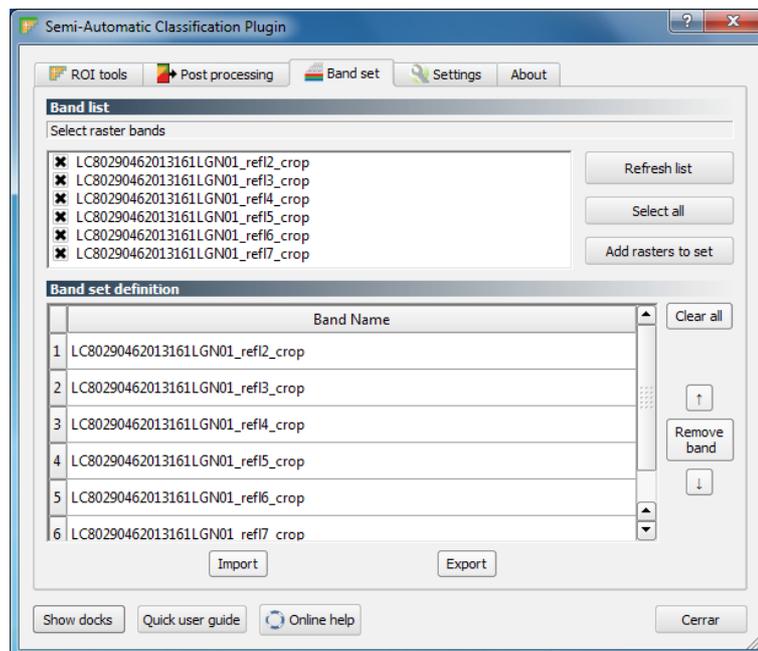


Figura 17: Bandas a utilizar en la creación de ROI en QGIS.

A continuación elegiremos las zonas de interés, para lo cual en '*creation*' en '*ROI parametres*' pulsamos en el botón "+". El cursor se convertirá en una cruz, de esta forma, mediante *click* en el centro de la región, se creará el ROI. Si la región obtenida no fue como deseábamos, podemos repetir la operación modificando los parámetros de la región.

Una vez tengamos una región adecuada, la guardaremos en el archivo '*shapefile*'. Para ello en el panel '*creation*' en '*ROI definition*' elegimos el identificador de su clase, '*ID*', su nombre de conjunto, '*Information*', y pulsamos el botón '*Save ROI*'. Para los conjuntos de entrenamiento seguiremos el siguiente esquema:

- ID: 1 = Agua Nombre conjunto : Water_X
- ID: 2 = Población Nombre conjunto : Poblacion_X
- ID: 3 = Cultivo Nombre conjunto : Farming_X
- ID: 4 = Bosque Nombre conjunto : Forest_X
- ID: 5 = Arena Nombre conjunto : Sand_X

Siendo X un número.

Repetiendo los anteriores pasos, creamos 5 ROI de cada clase. Obteniendo un total de 25 regiones, 5 de agua y 20 de tierra.

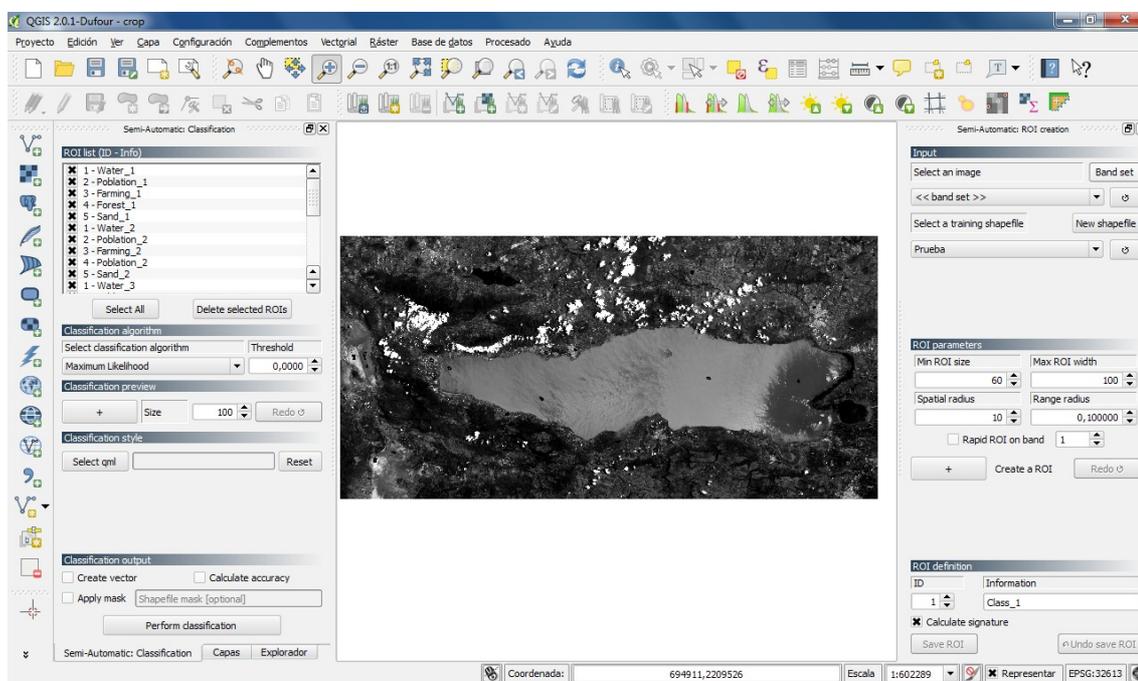


Figura 18: Regiones de interés en QGIS.

3.8 RASTERIZACIÓN DE LOS ROI

Una vez hayamos creado todas las regiones de interés, las exportaremos. QGIS mediante el *plugin* 'Semi-Automatic Classification' guarda los ROI en un archivo '*shapefile*' ('*.shp*'), pero al no tener constancia que este se encuentra en el estándar adecuado, exportaremos los ROI en otro formato '*shapefile*'.

Además tenemos otro problema. En los archivos '*.shp*' solo guarda las coordenadas de los puntos del perímetro de las regiones, y necesitamos las coordenadas de todos los puntos. Por lo tanto hemos de '*rasterizar*' los ROI. Este proceso lo realizaremos utilizando GRASS GIS.

Por lo tanto obtendremos dos archivos, un archivo '*.csv*' ('*comma-separated values*') con las coordenadas de los puntos (y la región a la que pertenece) y un archivo '*.shp*' con el resto de información. En resumen, realizaremos los siguientes pasos:

1. Exportaremos los ROI desde QGIS, el '*.shp*'.
2. '*Rasterizaremos*' los ROI en GRASS.

3. Exportaremos desde GRASS, el `'.csv'`.

3.8.1 Obtención del archivo `'.shp'`

Para realizar el primer paso, primero en el apartado capas, seleccionamos la capa de los ROI. Seguidamente accedemos a 'Capa -> Guardar como...' y seleccionamos las siguientes opciones:

- Formato: 'Archivo *shape* de *ESRI*'.
- Codificación: '*System*'.
- SRC: 'SRC seleccionado' y '*WGS 84/UTM zone 13N*'.

Así obtenemos el archivo `'.shp'`. Podemos observar cómo junto al archivo '*shapefile*', se crearon más archivos de diferentes formatos. Estos archivos son importantes, contienen información adicional. No los utilizaremos, pero han de estar junto al archivo '*shapefile*' (en la misma carpeta).

3.8.2 Obtención del archivo `'.csv'`

A continuación utilizaremos GRASS. Importamos el '*shapefile*' mediante '*Import vector data -> Common import formats*' (o comando `[v.in.ogr]`) como cualquier otro archivo. En la ventana de visualización podemos ver las regiones. Podemos añadir debajo una banda para comprobar cómo las regiones están georeferenciadas correctamente.

A continuación las regiones, que se encuentran en formato vector, las convertiremos en *ráster*. Este proceso lo hemos llamado '*rasterizar*'. De esta forma obtendremos las coordenadas de todos los puntos, de todas las regiones. Accedemos a '*Vector -> Map type conversions -> Vector to raster*' (o `[v.to.rast]`) y realizamos lo siguiente:

- 'Nombre del mapa vectorial de entrada:' Elegimos nuestro vector.
- 'Nombre del mapa ráster de salida': '*maskaROI*'.
- 'Entrada de valores ráster': '*cat*'.
- En la pestaña 'Selección', 'Tipo de objeto espacial': '*area*'.

Una vez hecho, podemos exportar las coordenadas de los puntos (junto a su número de conjunto). Nos dirigimos a '*File -> Export raster map -> ASCII x,y,z points export*' (o `[r.out.xyz]`).

- Nombre del archivo de salida: '*ROI**Sxyz.csv*'.
- Separador de campos: `';'`.

Así obtenemos el segundo archivo. Por lo tanto podemos pasar a la siguiente fase: la generación de la base de datos.

3.9 GENERACIÓN DE LA BASE DE DATOS

La generación de la base de datos se realizará mediante el lenguaje de programación R. El siguiente código (ver cuadrado 1) perteneciente al archivo 'principal.R' se encarga de dicha tarea. Mediante el análisis del código explicaremos como generamos la base de datos.

El programa comienza instalando una serie de librerías. Esta operación en R se realiza solo una vez. Tras esto carga las librerías, para poder operar con archivos tipo '*raster*'. Seguidamente leemos el archivo '*.shp*', el cual nos servirá para obtener la clase de nuestros ROI. Posteriormente leemos las posiciones de los ROI del archivo '*.csv*'. Usamos dos archivos '*.txt*', '*nombresBandas.txt*' y '*etiquetasBandas.txt*', los cuales contienen los nombres de las bandas (para facilitarnos su lectura) y el nombre (la etiqueta) que le daremos a la columna de cada banda.

A continuación leemos las bandas a utilizar, éstas serán de la banda 2 (Blue) a la 7 (SWIR2) y todas las operaciones entre bandas realizadas (apartado 2.3.2) siendo un total de 15 bandas. Éstas bandas las almacenaremos en una estructura tipo pila. Ésta estructura nos servirá para extraer toda la información de todas las bandas de todas las posiciones en las que se encuentran los 'ROI'. De esta estructura obtendremos una matriz ('matrizBandas') con una fila por posición y con 15 columnas correspondiente a su información en cada banda.

Esta matriz se convertirá en tipo '*dataframe*' y se le añadirá la clase. Así tendremos el '*dataframe*' 'datos5': la primera bases de datos. Además, a partir de 'datos5' obtendremos la base de datos 'datos2', la cual tendrá dos clases (1 y 2 correspondientes agua y tierra respectivamente) frente a las 5 clases de 'datos5' (1, 2, 3, 4 y 5 correspondientes a agua, población, cultivo, bosque y desierto respectivamente).

Por último exportamos las dos bases de datos en formato '*.csv*' para diversos usos.

```
# Instalar paquetes.
install.packages("raster")
install.packages("rgdal")
install.packages("maptools")

# Invocamos los paquetes o librerías.
library("raster")
library("rgdal")
library("maptools")

# Leemos los poligonos de entrenamiento del shapefile ('.shp') 'ROIS'.
r <- readShapePoly("ROIS/ROIS")
# Ver la variable.
```

```

r
# Ver la estructura de la variable.
str(r)

# Extraemos las clases y los conjuntos.
tabla <- as.data.frame(r@data)
# Leemos las posiciones de los puntos y el conjunto.
puntos <- read.csv(file="ROIs/ROISxyz.csv", header=F, sep=";")
# Extraemos las posiciones.
xy <- puntos[,1:2]
colnames(xy) <- c("x", "y")
# Extraemos los conjuntos.
nCjto <- puntos[,3]
# Obtenemos la clase de cada punto mediante el conjunto.
clase <- tabla$Primary.ID[nCjto]

# Leemos los nombres de las bandas del fichero 'nombresBandas.txt'.
nombresBandas = scan("nombresBandas.txt", what="character")
# Leemos las etiquetas de las bandas del fichero 'etiquetasBandas.txt'.
etiquetasBandas = scan("etiquetasBandas.txt", what="character")

# Creamos un grupo de bandas.
nBandas <- length(nombresBandas)
bandas <- raster(paste("bandas/",nombresBandas[1], ".TIF", sep=""))
for (i in 2:nBandas) {
  banda <- raster(paste("bandas/",nombresBandas[i], ".TIF", sep=""))
  bandas <- stack(bandas, banda)
}
nlayers(bandas)

# Extraemos las bandas del grupo de bandas correspondientes a las posiciones
# geográficas dadas por las coordenadas xy.
matrizBandas <- extract(bandas, xy)

# Vamos a dejar la matriz de los datos de entrenamiento en dos variables:
#   datos5 : un data.frame con 5 clases
#   datos2 : un data.frame con 2 clases: "agua" (1) y "no-agua" (2)

# Primero convertimos la matriz en 'data.frame'.
datos5 <- as.data.frame(matrizBandas)
# Le añadimos la clase.
datos5[, nBandas + 1] <- c(clase)
# Etiquetamos los nombres de las columnas.
colnames(datos5) <- c(etiquetasBandas, "clase")

# Primer 'data.frame' de entrenamiento.
write.table(datos5,file="datos5clases.csv", row.names=F, col.names=T, sep=",")

# Segundo 'data.frame' de entrenamiento.
datos2 <- datos5
datos2[which(datos2[,"clase"]>1),"clase"] <-2
write.table(datos2,file="datos2clases.csv", row.names=F, col.names=T, sep=",")

```

Algoritmo 1: Programa principal.

3.10 ENTRENAMIENTO DE CLASIFICADORES

El siguiente código es la primera parte del archivo 'training.R'. Esta primera parte se encarga del entrenamiento. No estamos entrenando los clasificadores definitivos (esto lo realizaremos en el siguiente apartado), estamos entrenando clasificadores para realizar la '*cross-validation*' empleando los mismos algoritmos. De este modo obtendremos los resultados que nos determinarán la calidad de los clasificadores.

Antes de ejecutar el código, R ha de tener en memoria los objetos creados en el apartado anterior. Porque usaremos las bases de datos 'datos2' y 'datos5' obtenidas previamente.

El programa actual empieza cargando las librerías y 'configurando' el 'Java' del ordenador. Posteriormente inicializamos las variables: números de '*fold*' a 2, veces que se realizará la '*cross-validation*' a 5 y la semilla a 1 (así la aleatoriedad tendrá siempre el mismo orden). Además, se convertirá la clase de las bases de datos a tipo factor para poder emplear las funciones de entrenamiento.

Continuamos entrenando la base de datos 'datos2'. Primero decidimos las opciones a seguir en el entrenamiento ('fitControl2'), después fijamos la semilla, entrenamos el primer clasificador, J48 y obtenemos los resultados. Repetimos el mismo proceso para los clasificadores NB y KNN.

Repetimos las ejecuciones realizadas con la base de datos 'datos2' con la base de datos 'datos5'. De este modo al ejecutar el código obtendremos los resultados de la '*cross-validation*' para los algoritmos J48, NB y KNN.

```
#install.packages("caret")
#install.packages("rJava")
#install.packages("pROC")
library("caret")
library(rJava) #Para J48
library(klaR) #Para NB
Sys.setenv(JAVA_HOME='C:\\Program Files (x86)\\Java\\jre7') #Para J48

#####
##Inicializaciones##
#####

datos5[, "clase"] <- factor(datos5[, "clase"])
datos2[, "clase"] <- factor(datos2[, "clase"])
testeo[, "clase"] <- factor(testeo[, "clase"])
Folds <- 2
Veces <- 5
Semilla <- 1

#####
##datos2##
#####

fitControl2 <- trainControl(method="repeatedcv", number=Folds, repeats=Veces,
classProbs=TRUE, summaryFunction=twoClassSummary)
```

```

# Tree - J48
set.seed(Semilla)
CV.train2.J48<-train(clase~.,data=datos2, method="J48", trControl=fitControl2)
CV.train2.J48
confusionMatrix(CV.train2.J48)

# NaiveBayes
set.seed(Semilla)
CV.train2.NB<- train(clase~., data=datos2, method="nb", trControl=fitControl2)
CV.train2.NB
confusionMatrix(CV.train2.NB)

# Knn
set.seed(Semilla)
CV.train2.KNN<-train(clase~.,data=datos2, method="knn", trControl=fitControl2)
CV.train2.KNN
confusionMatrix(CV.train2.KNN)

#####
##datos5##
#####

fitControl5 <- trainControl(method="repeatedcv", number=Folds, repeats=Veces,
classProbs=TRUE)

# Tree - J48
set.seed(Semilla)
CV.train5.J48<-train(clase~.,data=datos5, method="J48", trControl=fitControl5)
CV.train5.J48
confusionMatrix(CV.train5.J48)

# NaiveBayes
set.seed(Semilla)
CV.train5.NB<- train(clase~., data=datos5, method="nb", trControl=fitControl5)
CV.train5.NB
confusionMatrix(CV.train5.NB)

# Knn
set.seed(Semilla)
CV.train5.KNN<-train(clase~.,data=datos5, method="knn", trControl=fitControl5)
CV.train5.KNN
confusionMatrix(CV.train5.KNN)

```

Algoritmo 2: Primera parte del entrenamiento.

3.11 TESTEO DE CLASIFICADORES

Antes de realizar el proceso de test, hemos de repetir los apartados 3.7 y 3.8 ('Generación de los ROI desde Quantum GIS' y 'Rasterización de los ROI') para obtener un nuevo conjunto de regiones de interés (ROI) en un archivo '.shp' y en un archivo '.csv'. En esta ocasión a la hora de realizar el apartado 3.7, hemos de crear solo 2 clases: agua (clase 1) y tierra (clase 2). Con 10 regiones de agua y 10 regiones de tierra.

Posteriormente crearemos la base de datos para el conjunto de test con un programa muy similar al del apartado 3.9. Este programa corresponde al archivo 'test.R'. Para poder ejecutarlo hemos de tener en memoria los objetos creados en el apartado 3.9, dado que usaremos varios de esos objetos. A continuación analizaremos el código de 'test.R':

Cargamos las librerías necesarias. Leemos los archivos '.shp' y '.csv' correspondiente a las nuevas regiones de interés. Usando las bandas espectrales en memoria, extraemos la información correspondiente a las posiciones de los puntos en una matriz. Esta matriz la convertimos en tipo '*dataframe*', le añadimos la clase y la etiquetamos. De esta forma la base de datos del testeo será el objeto 'testeo'.

```
# Invocamos los paquetes o librerías
library("raster")
library("rgdal")
library("mapproj")

# Leemos los polígonos de entrenamiento del shapefile ('.shp') 'ROIs'.
r.test <- readShapePoly("ROIs/testeoROIs")
# Extraemos las clases y los conjuntos.
tabla.test <- as.data.frame(r.test@data)
# Leemos las posiciones de los puntos y el conjunto.
puntos.test <- read.csv(file="ROIs/testeoROIs.csv", header=F, sep=";")
# Extraemos las posiciones.
xy.test <- puntos.test[,1:2]
colnames(xy.test) <- c("x", "y")
# Extraemos los conjuntos.
nCjto.test <- puntos.test[,3]
# Obtenemos la clase de cada punto mediante el conjunto.
clase.test <- tabla.test$ID_class[nCjto.test]

# Extraemos las bandas del grupo de bandas correspondientes a las posiciones
# geográficas dadas por las coordenadas xy.
matrizBandas.test <- extract(bandas, xy.test)

# Primero convertimos la matriz en 'data.frame'.
testeo <- as.data.frame(matrizBandas.test)
# Le añadimos la clase.
testeo[, nBandas + 1] <- c(clase.test)
# Etiquetamos los nombres de las columnas.
colnames(testeo) <- c(etiquetasBandas, "clase")
```

Algoritmo 3: Preparación del conjunto de datos de test.

Posteriormente ejecutaremos la segunda parte del código del archivo 'training.R'. Éste se encargará de crear los clasificadores definitivos y de testarlos con el nuevo conjunto de datos. Previamente han de estar en memoria todos los objetos de 'principal.R', 'training.R' y 'test.R'.

En el código actual (cuadro 4) el primer paso será fijar la semilla. A continuación entrenaremos el clasificador, esta vez no incluirá opciones de control. De esta forma el clasificador es creado usando todo el conjunto de entrenamiento de 'datos2'. El clasificador obtenido será utilizado en la clasificación de toda la imagen posteriormente.

En el caso actual este clasificador clasificará el conjunto de testeo (objeto 'testeo'), y nos dará sus resultados. Finalmente guardaremos el objeto del clasificador obtenido como un archivo '.Rdata', para poder usarlo posteriormente. El proceso de entrenamiento y prueba será el mismo para los tres clasificadores (J48, NB y KNN). En el caso de J48 antes de guardarlo en formato '.Rdata', hemos de guardar el clasificador en la 'cache' de 'Java' para su correcto funcionamiento a posteriori.

```
#####
##testeo##
#####

# Tree - J48
set.seed(Semilla)
modelo.J48 <- train(clase~., data=datos2, method="J48")
testeo.J48 <- confusionMatrix(predict(modelo.J48, testeo), testeo[, "clase"])
testeo.J48
.jcache(modelo.J48$finalModel$classifier)
#save(modelo.J48, file="modelo.J48.RData")

# NaiveBayes
set.seed(Semilla)
modelo.NB <- train(clase~., data=datos2, method="nb")
testeo.NB <- confusionMatrix(predict(modelo.NB, testeo), testeo[, "clase"])
testeo.NB
#save(modelo.NB, file="modelo.NB.RData")

# Knn
set.seed(Semilla)
modelo.KNN <- train(clase~., data=datos2, method="knn")
testeo.KNN <- confusionMatrix(predict(modelo.KNN, testeo), testeo[, "clase"])
testeo.KNN
#save(modelo.KNN, file="modelo.KNN.RData")
```

Algoritmo 4: Entrenamiento de los clasificadores finales y realización del test.

Tras ejecutar este código obtenemos 3 clasificadores. Si observamos estos de cerca, podemos destacar ciertos aspectos:

- El algoritmo J48 solo emplea una banda. Creando un árbol con un único nivel con el valor 0.0595 como umbral. De este modo si una muestra es menor o igual a este valor clasificará la muestra como agua, si no como tierra.
- El algoritmo KNN toma el valor 9 para 'k'.

El motivo de empleo de una banda por el clasificador J48 es debido a que en esta banda el conjunto de entrenamiento es linealmente separable. Todos los valores mayores a cierto ancho de banda pertenecen a una clase y el resto a la otra clase. Esto podemos observarlo fácilmente en Weka.

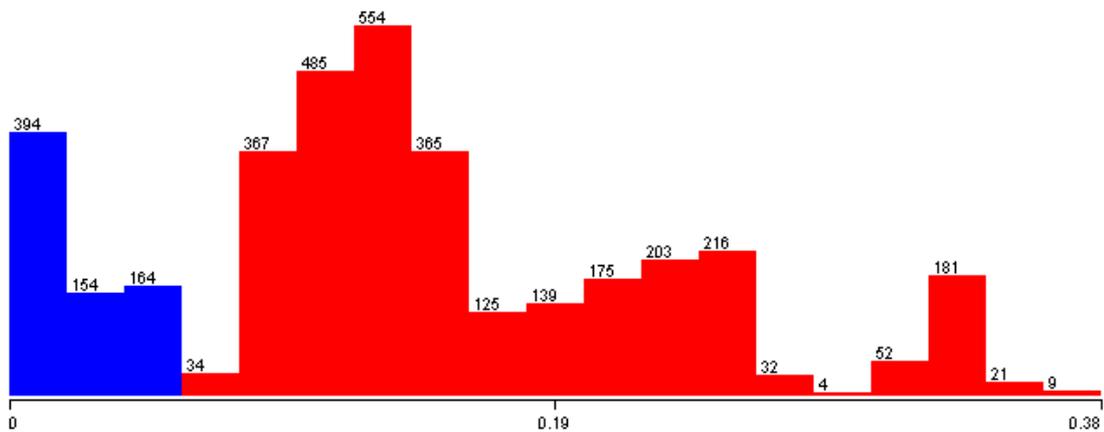


Figura 19: Conjunto de entrenamiento de banda SWIR2 en ancho de banda.

En la imagen podemos ver las muestras según su valor en la banda SWIR2. Siendo las muestras de color azul las de agua y las rojas las de tierra. Se puede apreciar como ambos conjuntos son disjuntos.

3.12 CLASIFICACIÓN

El siguiente paso será la clasificación de toda la imagen satelital. En esta tarea emplearemos GRASS y R conjuntamente, pero la versión de GRASS empleada será la 6.4.3, dado que faltan paquetes de R para poder funcionar completamente con la versión 7.0 de GRASS.

3.12.1 Creación de teselas

Una tesela será cada una de las piezas en las que dividiremos la imagen total. Nos servirá para poder clasificar la imagen por partes, porque clasificar toda la imagen a la vez sería muy costoso computacionalmente. Dividiéndola en teselas el coste computacional será mucho menor. Posteriormente empleando todas las teselas recompondremos la imagen resultado mediante la creación de un mosaico.

Empezaremos utilizando la línea de comandos de GRASS y ejecutando:

```
#####
## Dentro de GRASS##
#####
```

```
# Establecer la región.
g.region -d

# Llamada al programa R desde GRASS.
"C:\Program Files\R\R-3.0.1\bin\i386\Rgui.exe" &
```

Algoritmo 5: Llamada de GRASS a R.

Mediante el código anterior abrimos R desde GRASS. Dentro de R empezaremos situándonos en el directorio correcto. Después instalaremos los paquetes necesarios. El paquete 'sgrass6' nos permitirá trabajar con GRASS.

A continuación elegimos los parámetros de las teselas. Hemos elegido un tamaño de 256x256. Por lo tanto tendremos un total de 256 teselas.

Ejecutamos la función de GRASS 'r.tileset' para la creación de las teselas. Una vez creadas son ajustadas y finalmente guardadas en un archivo '.Rdata'.

```
#####
## Dentro de R##
#####

# Colocarnos justo en el directorio anterior donde estén las bandas.
setwd("C:/Users/Geingel/Documents/UPV/PFC/R")

#Instalar paquetes
install.packages("spgrass6", dependencies=T)
install.packages("stringr")
library(spgrass6)

# G <- gmeta6()

#####
## Creación de las teselas##
#####

# Parámetros de las teselas.
# Proyección = obtenida a partir de proyeccion <- projection(una_banda)
# Número máximo de columnas = maxC, Número máximo de filas = maxR
# Número de píxeles solapados en todas las direcciones = opixels
maxC = 256
maxR = 256
opixels = 3
proyeccion <- "+proj=utm +zone=13 +datum=WGS84 +units=m +no_defs +ellps=WGS84
+towgs84=0,0,0"

# Realizar.
coordenadas <- execGRASS("r.tileset", flags="g", parameters=list (sourceproj =
proyeccion, maxcols=maxC, maxrows=maxR, overlap=opixels), intern=TRUE)
lineas <- length(coordenadas)
coordenadas <- coordenadas[9:lineas]
coordenadas <- gsub(";", " ", coordenadas)

# Salvamos las coordenadas de las teselas.
save(coordenadas, file="coordenadasTeselas256.RData")
```

Algoritmo 6: Creación de las teselas.

Este código pertenece al archivo 'lanzadera_creacionTeselas_windows.R'.

3.12.2 Clasificación

Empezamos llamando a R desde la línea de comandos de GRASS, igual que en el paso anterior:

```
#####  
## Dentro de GRASS##  
#####  
  
# Establecer la región.  
g.region -d  
  
# Llamada al programa R desde GRASS.  
"C:\Program Files\R\R-3.0.2\bin\i386\Rgui.exe" &
```

Algoritmo 7: Llamada de GRASS a R.

Dentro de R empezamos situándonos en el directorio indicado. Seguidamente instalamos y cargamos las librerías necesarias para que el código funcione correctamente. A continuación leemos los nombres de las bandas, sus etiquetas y definimos el tipo de proyección. Posteriormente cargamos las teselas y el modelo de clasificación a usar. En el caso actual estamos empleando el clasificador J48, posteriormente hemos de repetir todo el proceso para los otros dos clasificadores. Seguidamente carga las funciones auxiliares mediante 'source' para su posterior uso.

Entonces llegamos al bucle. Éste se encarga del proceso de clasificación para cada trozo de la imagen, donde cada trozo es una tesela. En nuestro caso el bucle se ejecutará 256 veces. Se encargará de obtener la información de las bandas para cada tesela y con esta información clasificarla. De esta forma el bucle en cada iteración clasificará un trozo y escribirá el resultado en una imagen 'geoTIFF'. Empleando las imágenes obtenidas podremos realizar un mosaico y obtener la imagen binarizada completa (siguiente apartado).

```
#####  
## Dentro de R##  
#####  
  
# Colocarnos justo en el directorio anterior donde estén las bandas.  
setwd("C:/Users/Geingel/Documents/UPV/PFC/R")  
  
# Instalar librerías  
install.packages("spgrass6", dependencies=T)  
install.packages("RWeka")  
install.packages("e1071")  
install.packages("caret")  
install.packages("stringr")  
install.packages("raster")  
install.packages("klaR")
```

```

# Cargar librerías.
library(spgrass6)
library("RWeka")
library("e1071")
library("caret")
library("stringr")
library("raster")
library(klaR)

# G <- gmeta6()

#####
## Información de los nombres de bandas y etiquetas a utilizar##
#####

# Leemos los nombres de las bandas del fichero 'nombresBandas.txt'.
nombresBandas = scan("nombresBandas.txt", what="character")
nBandas <- length(nombresBandas)

# Leemos las etiquetas de las bandas del fichero 'etiquetasBandas.txt'.
etiquetasBandas = scan("etiquetasBandas.txt", what="character")

# Definimos el tipo de proyección.
proyeccion <- "+proj=utm +zone=13 +datum=WGS84 +units=m +no_defs +ellps=WGS84
+towgs84=0,0,0"

#####
## Clasificación de las teselas##
#####

load("coordenadasTeselas256.RData")
load("modelosClasificacion/modelo.J48.RData")

# Extracción de las teselas y su clasificación.
source("extraeWSEncr.R")
source("extraeBandasTile.R")
source("extraeMascaraTile.R")
source("clasificaTile.R")
nTiles <- length(coordenadas)

for (c in 1:nTiles) {
  message(sprintf("*** processing Tile: %d of %d...", c, nTiles))
  r <- extraeWSEncr(coordenadas[c])
  execGRASS("g.region", flag="g", parameters=list(w=r$w, s=r$s, e=r$e,
n=r$n, rows=as.numeric(r$rows), cols=as.numeric(r$cols)))
  mascara <- extraeMascaraTile()
  extMascara <- as(extent(mascara), "SpatialPolygons")
  crs(extMascara) <- proyeccion
  result <- extraeBandasTile(nombresBandas, etiquetasBandas, extMascara)

  # Hacer lo mismo para otros clasificadores.
  if (is.null(result))
    message(sprintf("NULL Tile: %d of %d...", c, nTiles))
  else
    clasificaTile(modelo.J48, as.data.frame(result$valores),
result$mapa, "modelo.J48")
}

```

3.12.3 Función auxiliar 1

Esta función auxiliar se encarga de extraer la información de una tesela. Esta información es relativa a su posición geográfica, de forma que el programa principal pueda llamar a '*g.region*' y situar geográficamente la tesela. Las variables '*w*', '*s*', '*e*' y '*n*' indican oeste (*west*), sur (*south*), este (*east*) y norte (*north*), mientras que '*cols*' y '*rows*' indican el número de columnas y de filas de la tesela.

```

extraeWSEncr <- function(x) {
  #x = coordenadas[k]
  w <- str_match(x, "w=[[[:digit:]]+[:punct:]][:digit:]]+")
  w <- str_match(w, "[[:digit:]]+[:punct:]][:digit:]]+")
  s <- str_match(x, "s=[[[:digit:]]+[:punct:]][:digit:]]+")
  s <- str_match(s, "[[:digit:]]+[:punct:]][:digit:]]+")
  e <- str_match(x, "e=[[[:digit:]]+[:punct:]][:digit:]]+")
  e <- str_match(e, "[[:digit:]]+[:punct:]][:digit:]]+")
  n <- str_match(x, "n=[[[:digit:]]+[:punct:]][:digit:]]+")
  n <- str_match(n, "[[:digit:]]+[:punct:]][:digit:]]+")
  cols <- str_match(x, "cols=[[[:digit:]]+>")
  cols <- str_match(cols, "[[:digit:]]+")
  rows <- str_match(x, "rows=[[[:digit:]]+>")
  rows <- str_match(rows, "[[:digit:]]+")

  result <- NULL
  result$w <- w
  result$s <- s
  result$e <- e
  result$n <- n
  result$cols <- cols
  result$rows <- rows

  return(result)
}

```

Algoritmo 9: Extracción de la información de las teselas.

3.12.4 Función auxiliar 2

La siguiente función crea un cuadrado a partir de la región previamente definida. Para posteriormente poder trabajar con una máscara. La cual se aplicará a las bandas para extraer la información de la tesela.

```

extraeMascaraTile <- function() {
  # Crea un grid a partir de la región.
  grd <- gmeta2grd()
  print(grd)
  ncells <- prod(slot(grd, "cells.dim"))
  df <- data.frame(k=rep(1, ncells))
}

```

```

mask_SG <- SpatialGridDataFrame(grd, data=df)
#print(summary(mask_SG))
return(mask_SG)
}

```

Algoritmo 10: Creación de máscara.

3.12.5 Función auxiliar 3

Dada la mascarada creada previamente, esta función se encarga de extraer la información correspondiente a la región de la máscara, para lo cual carga las bandas, les aplica la máscara y obtiene la región correspondiente. Si la matriz resultado es totalmente nula, lo cual se dará en los bordes de la imagen, devolverá nulo. En caso contrario, devolverá la información de las bandas para esa región.

```

extraeBandasTile <- function (nombresBandas, etiquetasBandas, extMascara) {
  flag <- 0
  nBandas <- length(etiquetasBandas)
  #message(sprintf("processing: band %d of %d...", 1, nBandas))
  valores <- NULL
  for (j in 1:nBandas) {
    #message(sprintf("processing: band %d of %d...", j, nBandas))
    banda <- raster(paste("bandas/",nombresBandas[j], ".TIF", sep=""))
    vBanda <- crop(banda, extMascara)
    idx <- which(vBanda[] != 0, arr.ind=TRUE)
    if (length(idx) == 0) {
      flag <- 1
      break
    }
    else {
      valores <- cbind(valores, vBanda[])
    }
  }
  result <- NULL
  if (flag == 0) {
    colnames(valores) <- etiquetasBandas
    mapa <- vBanda
    mapa[] <- 0
    result <- NULL
    result$mapa <- mapa
    result$valores <- valores
    rm(mapa)
  }
  rm(banda)
  rm(vBanda)
  return(result)
}

```

Algoritmo 11: Extracción de información de máscara.

3.12.6 Función auxiliar 4

La última función se encarga de clasificar la región actual. El resultado de la clasificación lo escribe en una imagen 'geoTIFF', junto a la información referencia a la posición de la tesela actual.

```
clasificaTile <- function (modelo, valores, mapa, idxModelo) {  
  mapa[] <- predict(modelo, as.data.frame(valores))  
  nombre <- paste("tiles/tile_", idxModelo, "_", c, ".tif", sep="")  
  writeRaster(mapa, filename=nombre, format="GTiff", datatype = "INT1U",  
  overwrite=TRUE)  
}
```

Algoritmo 12: Clasificación de tesela.

Tras realizar la clasificación al completo obtenemos una serie de imágenes, correspondientes a las teselas. Estas imágenes uniremos en un mosaico y visualizaremos los resultados finales. Obtendremos tres resultados correspondientes a los tres clasificadores empleados.

3.13 OBTENCIÓN DE RESULTADOS

El proceso de unión y visualización de resultados lo realizaremos mediante Quantum GIS. Para lo cual seguiremos los siguientes pasos:

1. Abrimos Quantum GIS y realizamos:
 - 'Ráster -> Miscelánea -> Índice de teselas'.
 - Seleccionamos el directorio de las teselas.
 - Elegimos un nombre para el archivo '.shp'.
2. Hemos creado el mosaico (figura 20). A continuación:
 - 'Ráster -> Miscelánea -> Construir ráster virtual (Catálogo)'.
 - Seleccionamos el archivo '.shp'.
 - Elegimos nombre para el archivo '.vrt'.
3. El mapa se muestra totalmente negro, entonces seleccionamos el mapa y le damos a 'Propiedades':
 - En 'Estilo' en 'Redenderizado de color':
 - MÍN = 0.
 - MÁX = 2.

De esta forma hemos unido los resultados de cada una de las teselas en un solo *ráster*. Así seremos capaces de visualizar el resultado de cada clasificación (ver siguiente capítulo).

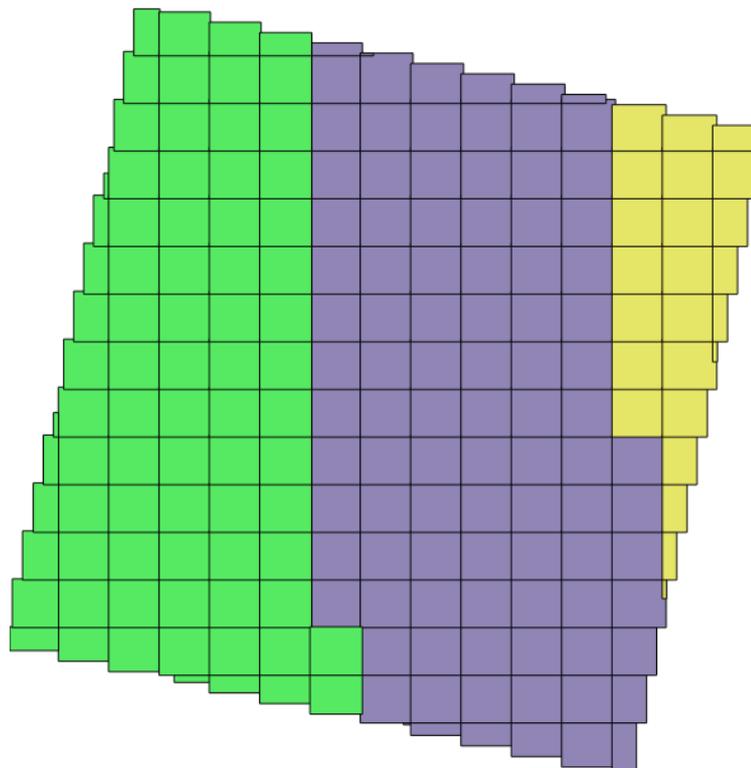


Figura 20: Mosaico creado a partir de teselas en QGIS.

3.14 COMPARACIÓN RESULTADOS Y NDWI

Hemos obtenido los resultados fruto de la clasificación. Pero para poder valorar la calidad de estos resultados hemos de tomar algún tipo de medida. La medida a tomar será la comparación de los tres resultados con los índices de agua (NDWI), para lo cual primero binarizaremos los índices de agua.

3.14.1 Umbrales para NDWI

Para binarizar cada NDWI hemos elegido un umbral. Este umbral ha sido calculado mediante procesos de clasificación. Esto es debido a las recomendaciones hechas por los autores de los umbrales [P7], los cuales generalmente recomiendan un umbral de 0 (u otro) y ajustar el umbral dinámicamente. Los umbrales lo hemos obtenido de la siguiente forma.

- Utilizando el software Weka.

- Emplearemos el conjunto de entrenamiento.
- Realizaremos el proceso de entrenamiento:
 - Con una única banda, la banda a umbralizar.
 - Utilizando el algoritmo J48.
 - Realizando '*cross-validation*'.

De este modo obtenemos un árbol con un único nivel, dado que solo se emplea una banda, el cual utilizando un umbral nos divide el conjunto de entrenamiento en dos. Este umbral será el correspondiente al índice de agua utilizado. Hemos obtenido los siguientes umbrales:

- NDWI1: 0.140418.
- NDWI2: -0.092539.
- NDWI3: -0.022651.
- NDWI4: -0.049356.

Como podemos observar son valores cercanos a cero, como recomiendan algunos autores[P7].

3.14.2 Binarización de NDWI

El proceso de binarización de los índices de vegetación será realizado en QGIS. Para realizar esta tarea, utilizaremos la calculadora *ráster*:

- 'Ráster → Calculadora ráster...'
- Escribimos una ecuación como la del siguiente ejemplo:
 - "NDWI4@1" <= -0.049356
- Escogemos la localización y el nombre.

De esta forma binarizamos los índices de agua. Repetiremos el mismo proceso para los resultados de la clasificación. Para tener ambas imágenes exactamente en el mismo rango de valores.

3.14.3 Resultados finales

Una vez binarizadas, tendremos 7 imágenes. 4 correspondientes a los NDWI y 3 a los resultados (J48, NB y KNN). A continuación realizaremos los siguientes pasos para obtener las comparaciones mediante la operación resta:

- Restaremos cada resultado (J48, NB y KNN) por cada NDWI como el ejemplo:
 - "J48@1_Binaria" - "NDWI4@1_Binaria"
- Analizaremos el histograma de cada resultado.
- Obtendremos la media y la desviación estándar de los resultados.

Por lo tanto tenemos 12 resultados, cada uno con su histograma, su media y su desviación estándar. Estos resultados se mostrarán en el siguiente capítulo.

'4'

Capítulo 4: Resultados

A continuación mostraremos los resultados obtenidos. Entre estos resultados se incluirán los resultados del entrenamiento, tanto para 2 clases como para 5, los resultados del test, los resultados de la clasificación y la comparación de los resultados finales con los índices de agua.

4.1 RESULTADOS ENTRENAMIENTO (2 CLASES)

A continuación mostraremos los resultados del entrenamiento:

	J48	NB FALSE	NB TRUE	KNN 5	KNN 7	KNN 9
ROC	0.998	1	1	1	1	1
Sensitivity	0.996	1	1	1	1	1
Specificity	1	1	1	1	1	1
ROC SD	0.00249	0	0	0	0	0
Sensitivity SD	0.00499	0	0	0	0	0
Specificity SD	0	0	0	0	0	0

Tabla 8: Resultados 'cross-validation' para 2 clases.

En la tabla 8 se dan los resultados mediante las siguientes medidas estadísticas: área bajo la curva ROC, sensibilidad, especificidad y las desviaciones estándares de estas. Se muestran los resultados para los 3 algoritmos de clasificación: J48, NB y KNN. Además, para el algoritmo NB disponemos resultados referentes a fallos y a aciertos. En cambio para el algoritmo KNN disponemos de resultados para distintos valores del parámetro 'k', siendo 'k' 5, 7 ó 9.

A continuación mostraremos las matrices de confusión obtenidas para los diferentes clasificadores. La 'Clase 1' corresponde a la clase de agua y la 'Clase 2' a la clase de tierra. Los valores de las tablas se encuentran en tanto por ciento del número total de muestras.

4.1.1 J48

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	19.3 %	0.0 %
Clase 2 predichos	0.1 %	80.6 %

Tabla 9: Matriz de confusión de J48 para 2 clases.

4.1.2 NB

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	19.4 %	0.0 %
Clase 2 predichos	0.0 %	80.6 %

Tabla 10: Matriz de confusión de NB para 2 clases.

4.1.3 KNN

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	19.4 %	0.0 %
Clase 2 predichos	0.0 %	80.6 %

Tabla 11: Matriz de confusión de KNN para 2 clases.

4.1.4 Conclusiones

Como hemos podido observar: Los resultados son prácticamente impecables. El porcentaje de acierto de los clasificadores NB y KNN es del 100%, mientras que para el algoritmo J48 es del 99.9%. Estos resultados tan buenos se deben principalmente a la gran diferencia de los atributos entre las muestras de agua y de tierra, lo cual las hace fácilmente distinguibles para los algoritmos siendo en algunos casos los datos linealmente separables.

4.2 RESULTADOS ENTRENAMIENTO (5 CLASES)

Repetimos el mismo proceso para 5 clases. En esta ocasión los estimadores empleados para valorar la calidad del entrenamiento serán la precisión (*Accuracy*) y coeficiente *kappa* de Cohen.

	J48	NB FALSE	NB TRUE	KNN 5	KNN 7	KNN 9
Accuracy	0.982	0.964	0.973	0.99	0.989	0.988
Kappa	0.978	0.954	0.966	0.987	0.986	0.985
Accuracy SD	0.00569	0.0039	0.00539	0.00207	0.00222	0.00156
Kappa SD	0.00717	0.00492	0.0068	0.00262	0.0028	0.00197

Tabla 12: Resultados 'cross-validation' para 5 clases.

A continuación mostraremos las matrices de confusión obtenidas. En este caso contamos con 5 clases: la 'Clase 1' corresponde a la clase de agua (*Water*), la 'Clase 2' a población (*Poblacion*), la 'Clase 3' a cultivo (*Farming*), la 'Clase 4' a bosque (*Forest*) y la 'Clase' 5 a desierto (*Sand*). Los valores se encuentran en tanto por ciento.

4.2.1 J48

	Clase 1 reales	Clase 2 reales	Clase 3 reales	Clase 4 reales	Clase 5 reales
Clase 1 predichos	19.3 %	0.0 %	0.0 %	0.0 %	0.0 %
Clase 2 predichos	0.1 %	19.2 %	0.0 %	0.0 %	0.3 %
Clase 3 predichos	0.0 %	0.1 %	12.9 %	0.2 %	0.0 %
Clase 4 predichos	0.0 %	0.0 %	0.2 %	23.0 %	0.0 %
Clase 5 predichos	0.0 %	0.7 %	0.2 %	0.0 %	23.8 %

Tabla 13: Matriz de confusión de J48 para 5 clases.

4.2.2 NB

	Clase 1 reales	Clase 2 reales	Clase 3 reales	Clase 4 reales	Clase 5 reales
Clase 1 predichos	19.4 %	0.0 %	0.0 %	0.0 %	0.0 %
Clase 2 predichos	0.0 %	19.1 %	0.1 %	0.0 %	1.3 %
Clase 3 predichos	0.0 %	0.0 %	12.9 %	0.0 %	0.0 %
Clase 4 predichos	0.0 %	0.0 %	0.3 %	23.1 %	0.0 %
Clase 5 predichos	0.0 %	0.9 %	0.0 %	0.0 %	22.9 %

Tabla 14: Matriz de confusión de NB para 5 clases.

4.2.3 KNN

	Clase 1 reales	Clase 2 reales	Clase 3 reales	Clase 4 reales	Clase 5 reales
Clase 1 predichos	19.4 %	0.0 %	0.0 %	0.0 %	0.0 %
Clase 2 predichos	0.0 %	19.5 %	0.1 %	0.0 %	0.0 %
Clase 3 predichos	0.0 %	0.1 %	12.9 %	0.0 %	0.0 %
Clase 4 predichos	0.0 %	0.0 %	0.1 %	23.2 %	0.0 %
Clase 5 predichos	0.0 %	0.5 %	0.2 %	0.0 %	24.1 %

Tabla 15: Matriz de confusión de KNN para 5 clases.

4.2.4 Conclusiones

En este caso podemos comprobar cómo los resultados al incluir las clases de los diferentes tipos de tierras empeoran, aunque no es un empeoramiento considerable, del 100-99.9% de acierto pasamos a un 99-96.4%. Además, en comparación con las matrices de confusión previas, el aumento del error se encuentra en los diferentes tipos de tierra ya que el porcentaje de acierto/error de agua y tierra sigue siendo el mismo.

4.3 RESULTADOS TESTEO

A la hora de presentar los resultados del testeo valoraremos la precisión (*Accuracy*), el coeficiente *kappa* de Cohen, la sensibilidad (*Sensitivity*) y la especificidad (*Specificity*).

En esta ocasión contamos con un mayor número de muestras dado que las regiones escogidas para el testeo son de mayor tamaño a las del entrenamiento. Así el testeo cuenta con

35675 muestras frente a 3674 en el caso del entrenamiento. Es decir, 10 veces mayor, en tamaño.

	J48	NB	KNN
Accuracy	0.9016	0.9991	1
Kappa	0.7983	0.9982	0.9999
Sensitivity	0.7858	0.9997	1.0000
Specificity	0.9989	0.9986	0.9999

Tabla 16: Resultados del testeo.

En el testeo volvemos a contar con 2 clases. 'Clase 1' como agua y 'Clase 2' como tierra. Mostraremos las matrices de confusión para los tres clasificadores.

4.3.1 J48

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	12799	21
Clase 2 predichos	3489	19366

Tabla 17: Matriz de confusión de J48 en el testeo.

4.3.2 NB

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	16283	27
Clase 2 predichos	5	19360

Tabla 18: Matriz de confusión de NB en el testeo.

4.3.3 KNN

	Clase 1 reales	Clase 2 reales
Clase 1 predichos	16288	1
Clase 2 predichos	0	19386

Tabla 19: Matriz de confusión de KNN en el testeo.

4.3.4 Conclusiones

Podemos observar cómo el algoritmo J48 empeoró significativamente, principalmente debido a que ciertas clases de agua las clasifica como tierra, como nos lo indica la sensibilidad. Esto puede ser debido a que, al usar distintas regiones de agua, de distintos lagos, no todas tienen las mismas propiedades, por poca profundidad, reflectancia del agua, u otros motivos y el clasificador ha tomado estas regiones como tierra. Aun así su precisión es del 90.16%, lo cual sigue siendo un buen índice de acierto.

Por otra parte, los clasificadores NB y KNN obtienen resultados increíbles con valores para los estimadores estadísticos del 99.9-100%. Estos resultados nos dan expectativas positivas de cara a una clasificación posterior.

A continuación, verificaremos si estos resultados tan buenos se mantienen a la hora de la clasificación de toda la imagen satelital multiespectral.

4.4 RESULTADOS CLASIFICACIÓN

Mediante la clasificación hemos obtenido tres imágenes binarias con valores 0 para agua y 1 para tierra. Mostraremos en una tabla la información de estas imágenes, usando como medidas: la cantidad de píxeles clasificados como agua, clasificados como tierra, la media y la desviación estándar sobre toda la imagen multiespectral. Nótese que la media y la desviación estándar se encuentran en el rango [0,1].

	J48	NB	KNN
Píxeles de agua	23744	16958	14316
Píxeles de tierra	452830	459616	462258
Media	0.95018	0.96442	0.96996
Desviación estándar	0.21758	0.18525	0.17070

Tabla 20: Resultados clasificación.

Las imágenes binarias se muestran a continuación. En estas el color negro corresponde a agua y el blanco a tierra.

4.4.1 J48

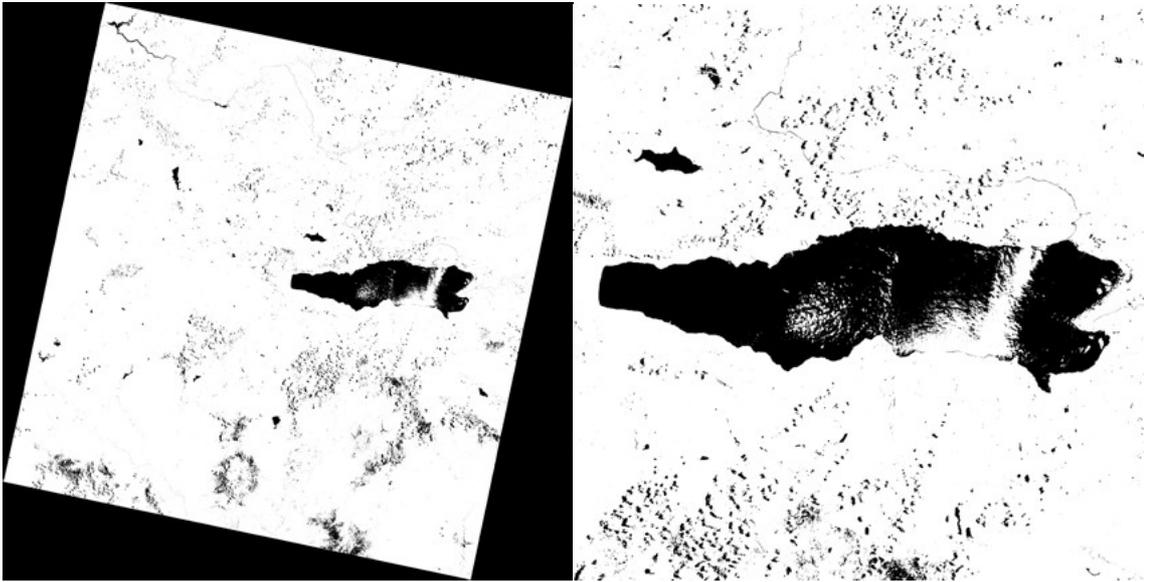


Figura 21: Resultados J48.

4.4.2 NB



Figura 22: Resultados NB.

4.4.3 KNN

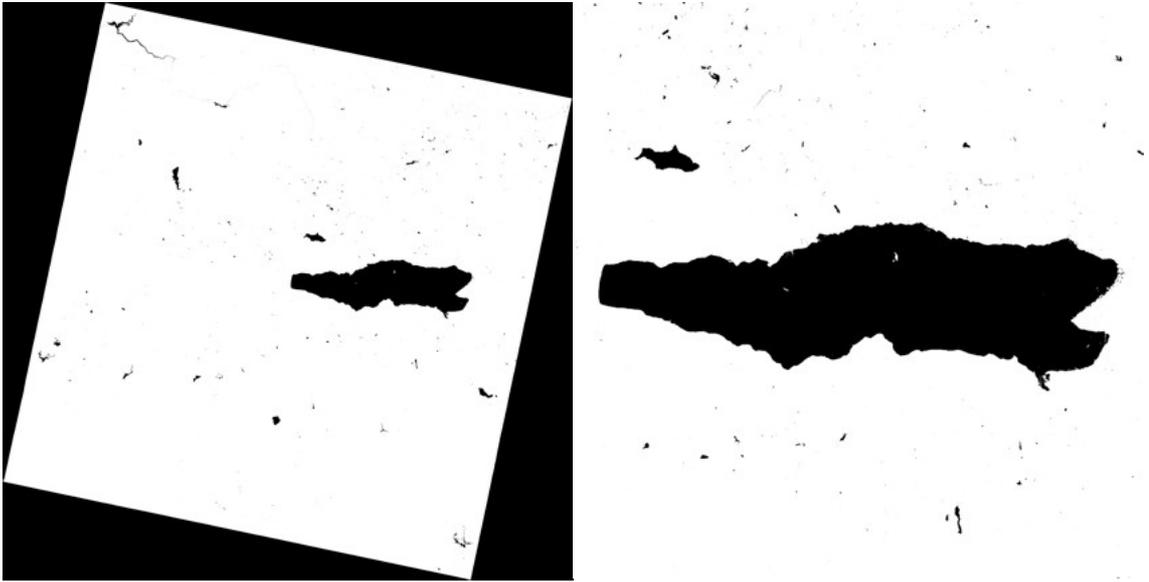


Figura 23: Resultados KNN.

4.4.4 Conclusiones

Analizando los resultados de forma visual podemos concluir que la distinción agua-tierra se ha realizado con mayor o menor éxito. Hemos comparado los tres resultados visualmente con la combinación RGB tierra/agua (sección 2.3.1).



Figura 24: Combinación RGB tierra/agua.

Podemos concluir lo siguiente:

- J48 no obtiene un buen resultado. Detecta los cuerpos de agua desde los lagos pequeños a los ríos, pero no clasifica el lago de Chapala correctamente. Además clasifica las sombras de las nubes como agua. Como sabemos J48 es un árbol de un único nivel que emplea la banda SWIR2. Al observar esta banda podemos ver que las sombras de las nubes tienen un tono muy oscuro, más oscuro que el lago principal. Podemos concluir que la elección del clasificador J48 al emplear sólo esta banda no fue una decisión muy acertada. Otra banda como la NIR o un índice de vegetación podría haber sido mejor opción.
- En el caso del clasificador NB tenemos un mejor resultado que para el algoritmo J48 ya que clasifica el lago principal correctamente y el resto de cuerpos de agua. No obstante, tiene problemas con las sombras de las nubes al igual que el algoritmo J48, ya que clasifica sombras de las nubes como agua, pero NB lo hace en menor medida.
- Podríamos decir que el clasificador KNN es la mejor elección dadas las comparaciones visuales ya que clasifica los cuerpos de agua correctamente y delimita correctamente el lago de Chapala. También tiene algunos problemas con las nubes; esto es, clasifica algunas sombras de las nubes como agua, pero en muy pequeña cantidad.

Si comparamos los resultados dados por la tabla 20 podemos concluir que J48 clasifica los cuerpos de agua por exceso (23744 píxeles de agua), mientras que KNN (14316 píxeles) lo hace por defecto y NB (16958 píxeles) se sitúa entre ambos.

Durante el análisis de resultados llegó a nuestro conocimiento, que sobre el Lago de Chapala crece lirio (una planta) en algunas zonas, debido a la contaminación de la zona y durante ciertas épocas del año. En los resultados obtenidos no se ha tenido este posible problema en cuenta pero podría ser una explicación para algunos de los resultados obtenidos.

4.5 COMPARACIÓN RESULTADOS Y NDWI

Hemos comparado cada resultado (J48, NB y KNN) con todas las bandas NDWI binarizadas mediante una resta de imágenes (véase apartado 3.14). Los resultados obtenidos los mostraremos en tres tablas, una por cada clasificador. Mediante distintas medidas obtenidas analizaremos la calidad de los resultados. Estas medidas serán:

- Número de coincidencias: Cantidad de píxeles iguales siendo ambos píxeles agua o ambos tierra. Corresponden al valor 0.
- Distancia de Hamming: Cantidad de píxeles diferentes siendo el primer píxel agua y el segundo tierra o viceversa.

- Agua (Clasificador) → Tierra (NDWI): Cantidad de píxeles donde el primer píxel es agua y el segundo tierra. Corresponden al valor -1.
- Tierra (Clasificador) → Agua (NDWI): Cantidad de píxeles donde el primer píxel es tierra y el segundo agua. Corresponden al valor 1.
- Media de los píxeles de la imagen diferencia: Se encuentra en el rango [-1,1].
- Desviación estándar de los píxeles de la imagen diferencia: Se encuentra en el rango [0,1].

Las primeras cuatro medidas han sido extraídas directamente de los histogramas.

4.5.1 J48

	NDWI1	NDWI2	NDWI3	NDWI4
Número de coincidencias	423977	461437	461578	463003
Distancia de Hamming	52536	15080	14936	13511
Agua (J48) → Tierra (NDWI)	3420	9435	8901	7072
Tierra (J48) → Agua (NDWI)	49116	5645	6035	6439
Media	0.09590	-0.00795	-0.00601	-0.00133
Desviación estándar	0.31789	0.17772	0.17694	0.1683

Tabla 21: Comparación resultados J48 y NDWI.

4.5.2 NB

	NDWI1	NDWI2	NDWI3	NDWI4
Número de coincidencias	419235	469729	470538	469407
Distancia de Hamming	57278	6788	5976	7107
Agua (NB) → Tierra (NDWI)	2398	1896	1028	477
Tierra (NB) → Agua (NDWI)	54880	4892	4948	6630
Media	0.11014	0.00629	0.00823	0.01291
Desviación estándar	0.32874	0.11919	0.11168	0.12144

Tabla 22: Comparación resultados NB y NDWI.

4.5.3 KNN

	NDWI1	NDWI2	NDWI3	NDWI4
Número de coincidencias	417131	470671	469854	467645
Distancia de Hamming	59382	5846	6660	8869
Agua (KNN) → Tierra (NDWI)	2129	104	49	37
Tierra (KNN) → Agua (NDWI)	57253	5742	6611	8832
Media	0.11568	0.01183	0.01377	0.01846
Desviación estándar	0.33352	0.11013	0.11742	0.13517

Tabla 23: Comparación resultados KNN y NDWI.

4.5.4 Conclusiones

Observando los resultados de las tablas 21, 22 y 23 podemos concluir lo siguiente:

- El índice NDWI1 no parece ser un buen índice ya que no permite una buena binarización. Hemos analizado varios umbrales gráficamente, pero ninguno dio un resultado satisfactorio puesto que clasificaba muchas zonas de tierra como agua y, ni siquiera, clasificaba el lago de Chapala correctamente. Por ello observamos que en las tablas anteriores, en la medida 'Tierra (Clasificador) → Agua (NDWI)', el índice NDWI1 muestra una cantidad cercana a 50000 píxeles, mientras que, para el resto de índices de agua, el número de píxeles es cercano a 6000.
- J48 clasifica los cuerpos de agua por exceso al ser el único de los algoritmos que presenta medias negativas con respecto a los índices de agua. Además, la distancia de Hamming y sus desviaciones estándares son superiores a las obtenidas por los algoritmos NB y KNN lo cual nos indica que es el resultado con menor parecido a los índices de agua, en comparación con los restantes algoritmos de clasificación.
- Los tres clasificadores tienen una cantidad similar en la medida 'Tierra (Clasificador) → Agua (NDWI)' rondando los 6000 píxeles, lo cual puede indicar dos cosas: hay zonas de agua que los tres clasificadores no detectan o los índices de agua clasifican ciertas zonas de tierra como agua lo cual parece bastante improbable. Es por ello que nos inclinamos a pensar que muchas zonas húmedas no son detectadas por los algoritmos y sí por los índices de agua y que la detección de los algoritmos se centra, más bien, en los cuerpos de agua.

- El algoritmo NB y, sobre todo, el algoritmo KNN clasifican por defecto ya que la medición 'Agua (Clasificador) → Tierra (NDWI)' así nos lo indica con un número de píxeles menores a 1897 en el caso del algoritmo NB y menores a 105, en el caso del algoritmo KNN (si exceptuamos el caso del índice de agua NDWI1).

De esta forma podemos concluir que el mejor resultado, en nuestro caso, es el obtenido por el algoritmo de clasificación KNN seguido del resultado del algoritmo NB. También podemos concluir que el clasificador J48 parece ser una mala opción para la clasificación de la imagen satelital multiespectral utilizada en la detección de cuerpos de agua.

'5'

Capítulo 5: Conclusiones y líneas futuras

En este capítulo trataremos sobre las conclusiones obtenidas del proyecto, así como posibles líneas futuras a seguir sobre el mismo.

Recordemos cómo el objetivo principal fue definido como la distinción de agua y tierra de una imagen satelital multiespectral. Partiendo de la imagen satelital, mediante una serie de procesos, hemos llegado a realizar esa distinción con éxito. En los resultados podemos apreciar cómo el lago principal y otros cuerpos de agua más pequeños son distinguidos de la tierra.

Por lo tanto, el objetivo principal y todos los demás objetivos han sido cumplidos con éxito.

Dentro del tratamiento de imágenes satelitales existen grandes posibilidades. Hemos utilizado algunas de ellas, empleando ciertos *software* tipo GIS y un lenguaje de programación de alto nivel. Por lo tanto, empleando las herramientas adecuadas y realizando cierto número de pasos hemos cumplido todos nuestros objetivos.

Dado que hemos trabajado con una imagen real podríamos decir que el proyecto está cerca de un caso real, pero si analizamos mejor todos los factores comprobaremos que no es cierto. Hemos empleado una imagen real, pero solamente una. Esta imagen fue escogida minuciosamente, asegurándose de tener el mínimo número de elementos ruidosos, como nubes o nieve. Empleando varias imágenes en vez de una añadiríamos muchas complicaciones al proceso de distinción de cuerpos de agua localizados en distintos sitios del mundo, en

distintas épocas del año, bajo diferentes efectos atmosféricos, a partir de distintos satélites, con o sin contaminación del entorno,...

Por lo tanto, en un caso real el proceso de distinción sería mucho más complicado, teniendo que lidiar con más variables o factores. Solucionar todas estas complicaciones y crear un proceso (o varios) más general podría ser una línea futura de trabajo. Empezando por un caso más inmediato, como emplear diferentes imágenes del mismo sitio geográfico de diferentes épocas del año y ver cómo se comporta el proceso actual ante ellas y evaluar posibles mejoras.

A la hora de evaluar los resultados finales comentamos el descubrimiento de lirio en el lago de Chapala. Esto ha hecho que los resultados no fueran del todo correctos. Sería interesante en un futuro, solucionar este problema. Analizar distintas imágenes teniendo en cuenta esta planta y cómo evoluciona a lo largo del año para así obtener unos resultados más realistas.

Otra posible mejora del proceso sería la paralelización del código de clasificación. La clasificación es muy costosa, dado que trabajamos con grandes volúmenes de datos. Teniendo en cuenta que el equipo de trabajo empleado contaba con dos núcleos, la paralelización no aportaba gran mejora a la velocidad de cálculo, por lo tanto se desestimo paralelizar el proceso. Pero en un caso real, la utilización de mejores unidades de cómputo, sería algo imprescindible. Todavía más importante si empleamos más de una imagen.

Una última posible aspiración sería la conversión del proceso actual en automático. Para poder llegar a cumplir este nuevo reto, la mejor opción sería la inclusión de todo el proceso en un *software* GIS a modo de *plugin*. A la hora de la creación de las regiones de interés (ROI) en QGIS vimos el *plugin* '*Semi-Automatic Classification*'. Este además de ayudarnos en el proceso de obtención de los ROI, disponía de las posibilidades de clasificación empleando los ROI. Al ser estas posibilidades muy limitadas se descartó su uso, pero esto nos da una visión de cómo podría llegar a incluirse nuestro proceso en un *software* GIS. De forma similar podríamos convertir nuestro proceso en automático.

'6'

Capítulo 6: Bibliografía

- [B1] Kevin Butler. Band Combinations for Landsat 8, 24-07-2013.
<http://blogs.esri.com/esri/arcgis/2013/07/24/band-combinations-for-landsat-8/>
- [B2] Texas Natural Resources Information System. Landsat 8 Satellite Imagery Now Available, 20-06-2013. <http://www.tnris.org/landsat-8-satellite-imagery-now-available>
- [B3] Franz Leonardo. Combinación de bandas para Landsat 8, 28-06-2013.
<http://acolita.com/combinacion-de-bandas-para-landsat-8/>
- [B4] U.S. Geological Survey. Frequently Asked Questions about the Landsat Missions, 07-12-2013 http://landsat.usgs.gov/L8_band_combos.php
- [B5] Charlie Loyd. Putting Landsat 8's Bands to Work, 14-06-2013.
<https://www.mapbox.com/blog/putting-landsat-8-bands-to-work/>
- [B6] Franz Leonardo. Composición de bandas satelitales en ArcGIS, 22-05-2013.
<http://acolita.com/composicion-de-bandas-satelitales-landsat-etm-en-arcgis/>
- [C1] Colaboradores Wikipedia. Wikipedia: C4.5, 16-07-2013
<http://es.wikipedia.org/wiki/C4.5>

- [C2] Colaboradores Wikipedia. Wikipedia: Clasificador bayesiano ingenuo, 16-06-2014. http://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo
- [C3] Colaboradores Wikipedia. Wikipedia: Knn, 09-01-2014. <http://es.wikipedia.org/wiki/Knn>
- [C4] Bruno López Takeyas. Algoritmo C4.5, 2005. Nuevo Laredo, Tamaulipas. [http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5\(2005-II-B\).pdf](http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5(2005-II-B).pdf)
- [G1] Markus Neteler. Processing Landsat 8 data in GRASS GIS 7: Import and visualization, 29-09-2013. <http://courses.neteler.org/processing-landsat8-data-in-grass-gis-7/>
- [G2] Markus Neteler. Processing Landsat 8 data in GRASS GIS 7: RGB composites and pan sharpening, 05-10-2013. <http://courses.neteler.org/processing-landsat-8-data-in-grass-gis-7-rgb-composites-and-pan-sharpening/>
- [G3] Landmap Geoknowledge. Calculating Vegetation Indices, 2013. <http://landmap.mimas.ac.uk/index.php/Learning-Materials/Image-Processing-for-GRASS/5.2-Calculating-Vegetation-Indices>
- [L1] EoPortal Directory. Landsat-8 / LDCM (Landsat Data Continuity Mission), 2013 <https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-8-ldcm>
- [L2] U.S. Geological Survey. Landsat 8, 03-5-14 <http://landsat.usgs.gov/landsat8.php>
- [L3] Mapping & Co. Disfrutando del Landsat 8 (1 parte – Especificaciones técnicas), 21-08-2013. <http://mappingandco.com/blog/disfrutando-del-landsat-8-1-parte-especificaciones-tecnicas/>
- [L4] Dr. José Ramón Martínez Batlle. Lanzamiento del satélite LDCM (Landsat Data Continuity Mission), 11/02/2013. <http://www.geografiafisica.org/2013/02/11/lanzamiento-del-satelite-ldcm-landsat-data-continuity-mission/>
- [L5] NASA. LDCM Press Kit, febrero de 2013. http://www.nasa.gov/pdf/723395main_LDCMpresskit2013-final.pdf
- [L6] Centro de Artigos. Programa Landsat, Historia, Satélite cronología, Detalles técnicos, Futuro, 28-03-2014 http://centrodeartigos.com/articulos-enciclopedicos/article_80378.html
- [L7] Colaboradores Wikipedia. Wikipedia: Landsat 8, 2014 http://en.wikipedia.org/wiki/Landsat_8

- [P1] Alexander Ariza. Descripción y Corrección de Productos Landsat 8 LDCM (Landsat Data Continuity Mission) Versión 1.0, 2013. Instituto Geográfico Agustín Codazzi.
- [P2] Luca Congedo. Semi-Automatic Classification Plugin for QGIS User Manual Version 1.0.0. octubre 2013.
- [P3] Ouma, Y.O., and R. Tateishi, 2006. A water index for rapid mapping of shoreline changes of five East African Rift Valley lakes: An empirical analysis using Landsat TM and ETM+ data, *International Journal of Remote Sensing*, 27(15):3153–3181.
- [P4] Gao B., 1996. NDWI – A normalized difference water index for remote sensing of vegetation liquid water from space, *Remote Sensing of Environment*, 58(3):257–266.
- [P5] McFeeters, S.K., 1996. The use of Normalized Difference Water Index (NDWI) in the delineation of open water features, *International Journal of Remote Sensing*, 17(7):1425–1432.
- [P6] Rogers, A.S., and M.S. Kearney, 2004. Reducing signature variability in unmixing coastal marsh Thematic Mapper scenes using spectral indices, *International Journal of Remote Sensing*, 25(12):2317–2335.
- [P7] Ji, L., Zhang, L., & Wylie, B., 2009. Analysis of dynamic of thresholds for the Normalized Difference Water Index. *Photogrammetric Engineering and Remote Sensing*, 75(11), 1307–1317.
- [P8] Zahidul Islam, 2007. Determination of flood extent using remote sensing. Technical report, Department of Civil and Environmental Engineering.
- [P9] Xu, H., 2006. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery, *International Journal of Remote Sensing*, 27(14):3025–3033.
- [Q1] Luca Congedo. Install the Semi-Automatic Classification Plugin in QGIS 2: a Windows guide, 28-09-2013
<http://fromgistors.blogspot.com.es/2013/09/install-semi-automatic-classification-windows.html>
- [Q2] Luca Congedo. Exploring the Features of the Semi-Automatic Classification Plugin for QGIS, Version 2.1.0, 27-08-2013.
<http://fromgistors.blogspot.com.es/2013/08/features-semi-automatic-classification-plugin-2.1.0.html>

- [R1] David V. Conesa Guillén. Curso de Introducción al entorno R, 2014. <http://www.uv.es/conesa/CursoR/cursor.html>
- [R2] Robert J. Hijmans. Introduction to the 'raster' package (version 2.2-31), 6-03-2014. <http://cran.r-project.org/web/packages/raster/vignettes/Raster.pdf>
- [R3] Kurt Hornik. Package 'RWeka', 26-04-2014. <http://cran.r-project.org/web/packages/RWeka/RWeka.pdf>
- [R4] Max Kuhn. Package 'caret', 4-06-2014. <http://cran.r-project.org/web/packages/caret/caret.pdf>
- [R5] Josh Walters. Naive Bayes Classification in R, 27-11-2012. <http://joshwalters.com/2012/11/27/naive-bayes-classification-in-r.html>
- [R6] Caret. Model training and tuning, 31-05-2014. <http://caret.r-forge.r-project.org/training.html>
- [R7] Max Kuhn, Ph.D. Predictive Modeling with R and the caret Package useR!, 2013. http://www.edii.uclm.es/~user-2013/Tutorials/kuhn/user_caret_2up.pdf

