



FACULTAD DE INFORMÁTICA.  
UNIVERSIDAD COMPLUTENSE DE MADRID.

# Clasificación de texturas naturales mediante técnicas de visión por computador en imágenes aéreas.

---

Sistemas informáticos  
[Curso 2008-09]

Alumnos:

Elena Fernández Albillos.  
Luis Lópiz Morales.  
Fernando Yubero Yubero.

Profesor director:

Gonzalo Pajares Martinsanz.

# ÍNDICE

<b>1. INTRODUCCIÓN.....</b>	<b>5</b>
1.1 JUSTIFICACIÓN DEL PROYECTO.....	5
1.2 PLANIFICACIÓN DEL PROYECTO.....	6
<b>2. PLANIFICACIÓN.....</b>	<b>9</b>
2.1 PLAN DE FASE.....	9
2.1.1 FASE DE INICIO.....	9
2.1.2 FASE DE ELABORACIÓN.....	10
2.1.3 FASE DE CONSTRUCCIÓN.....	10
2.1.3.1 Iteración 1.....	10
2.1.3.2 Iteración 2.....	11
2.1.3.3 Iteración 3.....	11
2.1.4 FASE DE TRANSICIÓN.....	12
<b>3. ANÁLISIS.....</b>	<b>13</b>
3.1 ESPECIFICACIÓN DEL PROYECTO.....	13
3.1.1 ALGORITMO DE CUANTIZACIÓN VECTORIAL.....	13
3.1.2 ALGORITMO DE LLOYD.....	13
3.1.3 ALGORITMO DE AGRUPAMIENTO BORROSO.....	15
3.1.4 ALGORITMO BAYESIANO.....	18
3.2 REQUISITOS.....	21
3.2.1 REQUISITOS FUNCIONALES.....	22
3.2.2 REQUISITOS NO FUNCIONALES.....	22
3.3 RIESGOS.....	22
3.3.1 TECNOLÓGICOS.....	23
3.3.2 DE PERSONAL.....	24
3.3.3 ORGANIZATIVOS.....	25
<b>4. DESARROLLO.....</b>	<b>27</b>
4.1 FASES DE DESARROLLO.....	27
4.1.1 FASE DE INICIO.....	27
4.1.2 FASE DE ELABORACIÓN.....	27
4.1.3 FASE DE CONSTRUCCIÓN.....	28
4.1.4 FASE DE TRANSICIÓN.....	30
4.2 PLAN DE PRUEBAS.....	30
<b>5. DISEÑO.....</b>	<b>35</b>
5.1 ESPECIFICACIÓN DEL DISEÑO.....	35
5.2 DIAGRAMAS UML.....	36
5.3 DIAGRAMAS DE SECUENCIA.....	38

<b>6. RESULTADOS PROYECTO.....</b>	<b>41</b>
6.1 EJECUCIÓN 1.....	41
6.2 EJECUCIÓN 2.....	44
<b>7. BIBLIOGRAFÍA.....</b>	<b>48</b>
<b>8. ANEXO.....</b>	<b>49</b>

## RESUMEN

El sistema implementado consiste en una aplicación que realiza una clasificación automática de texturas en imágenes digitales a partir de un entrenamiento previo no supervisado.

El algoritmo elegido para el entrenamiento no supervisado es el conocido como cuantización vectorial.

Para la clasificación de las diferentes texturas se han implementado los tres algoritmos siguientes: Algoritmo de Bayes, Algoritmo de clasificación Fuzzy (borroso) y Algoritmo de Lloyd

Para facilitar su utilización, se ha dotado al sistema de interfaz gráfica de usuario desde la cual se puede elegir la operación a realizar y configurar los parámetros necesarios.

La aplicación ha sido desarrollada en Java utilizando la librería Java Advanced Image bajo la plataforma Eclipse.

**Palabras Clave:** Clasificación, Entrenamiento, Algoritmo, Agrupamiento Borroso, Clasificador de Bayes, algoritmo de Lloyd, Cuantización vectorial, procesamiento de imágenes

## ABSTRACT

The implemented system consists of an application that performs automatic textures classification in digital images starting from a previous unsupervised learning approach.

The chosen algorithm for the learning phase is the well known unsupervised learning vector quantization.

For classifying the different textures in the images, the following three algorithms have been developed: Bayesian, Fuzzy clustering and Lloyd's.

For an easy use, the system has been equipped with an interface from which the user is able to choose the operation to perform and configure the necessary parameters.

The application has been developed in the Java language, using the Java Advanced Image library, under the Eclipse platform.

**Key Words:** Classification, Training, Algorithm, Fuzzy clustering, Bayesian classifier, Lloyd's algorithm, Learning vector quantization, image processing

## AUTORIZACIÓN

Autorizamos a la facultad de Informática de la Universidad Complutense de Madrid, así como al resto de los centros adscritos a la Universidad Complutense de Madrid a la utilización y modificación de todos los componentes que forman este proyecto. Siempre y cuando no se derive en una utilización comercial de cualquiera de los componentes o de parte de ellos.

Firmado:

.....  
Luis Lópiz Morales

.....  
Fernando Yubero Yubero

.....  
Elena Fernández Albillos

# 1. INTRODUCCIÓN

## 1.1 JUSTIFICACIÓN DEL PROYECTO

Actualmente existe un creciente interés en el desarrollo de procedimientos para el tratamiento de las texturas, siendo la clasificación un tema de especial relevancia.

Este interés tiene su fundamento en algunos aspectos tales como:

1. Control de cultivos en agricultura, propiciado por la necesidad de conocer los cultivos programados para la recepción de subvenciones u otros aspectos relacionados.
2. Cómputo y medición de parcelas agrícolas y tipo de cultivos a los que se dedica.
3. Control de riegos agrícolas
4. Agricultura de precisión para aplicar herbicida en el tratamiento de malas hierbas de forma selectiva evitando la contaminación medioambiental y la reducción de costes de producción.
5. Evaluación de catástrofes naturales: fuegos, daños por inundaciones, heladas en cultivos agrícolas, nevadas, etc.
6. Detección de cambios en determinadas zonas, principalmente urbanas para el control de edificaciones o impactos medioambientales
7. Vigilancia en prevención de catástrofes, por ejemplo fuegos o inundaciones.
8. Control de fenómenos meteorológicos como es el retroceso de determinadas playas.
9. Vigilancia: forestal, marítima.
10. Detección de infraestructuras: carreteras, caminos forestales, cañadas reales, etc.

En este sentido, diversas empresas u organismos desarrollan o utilizan aplicaciones para abordar dicha problemática. Por citar sólo algunas podemos mencionar

- 1) Digital Image Processing (Dimap) (<http://www.dimap.es/>). Las imágenes utilizadas en el proyecto son cortesía del Servicio Territorial de Galicia (SITGA) proporcionadas por Dimap y pertenecientes a la región de Abadín (Lugo) capturadas mediante vuelos aéreos.
- 2) Proespacio (<http://www.proespacio.org/>) agrupación de empresas del sector aeroespacial donde algunas de las actividades destacables son aplicaciones mediante el uso de imágenes de satélite. En este consorcio destacan algunas empresas líderes del sector tanto en España como a nivel Internacional:

EADS-Espacio, EADS Astrium, CRISA, GMV, Indra Espacio, Sener, Hispasat, IberEspacio, Inasmet, Insa, Mier, NTE, Tecnológica, Ryma, Hispasat, GTD, Alcatel, CRISA, GTD.

### 3) Organismos oficiales y Centros de Investigación

- Consejo Superior de Investigaciones Científicas (CSIC)
- Instituto Nacional de Técnica Aeroespacial (INTA)
- Centro de Estudios y Experimentación de Obras Públicas (CEDEX), con el que existen trabajos de colaboración previos por parte de uno de los directores del trabajo (Pajares y col. 2001, Pajares y col. 2002)

La mayoría de las empresas e instituciones anteriormente mencionadas utilizan para llevar a cabo sus aplicaciones principalmente herramientas comerciales tales como:

- ERDAS
- Intergraph
- ENVI-IDL
- ILOG
- PCI
- E-Cognition

Aunque bien es cierto que cada día es mayor la potencialidad de las herramientas mencionadas no es menos cierto que los retos tecnológicos derivados de las aplicaciones anteriores hacen que en algunos casos la utilización de tales herramientas sea insuficiente para abordar las propuestas de proyectos demandados por los clientes. Este es el caso para una gran parte de las aplicaciones mencionadas en el tratamiento de las texturas naturales donde la **clasificación** de las mismas surge como una tarea fundamental.

En efecto, en general las mencionadas herramientas implementan algunos de los métodos clásicos de clasificación, siendo necesaria la intervención del usuario mediante programación para abordar algunas de las tareas que involucran aspectos de clasificación de texturas, lo cual no siempre es factible o al menos en la medida que cabría esperar debido a múltiples limitaciones.

Además, y lo que es más importante, en muchos casos no existe la posibilidad de llevar a cabo la investigación necesaria para abordar la problemática, particularmente cuando los métodos clásicos no producen los resultados esperados.

Por todo lo expuesto anteriormente, surge una necesidad importante en el ámbito de las aplicaciones reales para abordar el tema de la clasificación de texturas naturales en imágenes y un reto para la comunidad científica para tratar de mejorar los procedimientos existentes con la mayor flexibilidad posible.

## 1.2 PLANIFICACIÓN DEL PROYECTO

El proyecto se organiza de forma que en la sección dos se determina la planificación inicial del proyecto, distribuyéndose las tareas entre los miembros del equipo y realizando una serie de actividades en forma de iteraciones encaminadas a definir y

redefinir las actividades en función de las necesidades o dificultades identificadas en cada etapa.

En la sección tres se realiza el análisis preliminar del proyecto, planteándose los métodos y algoritmos a implementar; se identifican y definen simultáneamente los requisitos del sistema. Se avanza también los posibles riesgos tecnológicos identificados hasta el momento.

En la sección cuatro se especifican y definen las fases de desarrollo, así como el plan de pruebas previsto.

En la sección cinco, se propone un diseño detallado basado en el paradigma UML, identificándose los módulos, clases y los diagramas correspondientes bajo esta metodología.

En la sección seis se muestran los resultados más relevantes del proyecto.

Finalmente, se incluye un Anexo, que contiene el manual de usuario de la aplicación.



## 2. PLANIFICACIÓN

Por planificación se entienden todos aquellos aspectos relacionados tanto con la organización del proyecto como con la estructuración de los diferentes planes previstos. A continuación se detallan ambos aspectos.

### 2.1 ORGANIZACIÓN DEL PROYECTO.

El proyecto ha sido realizado por un grupo de tres alumnos, de forma que cada miembro se encargara de uno de los algoritmos elegidos para la implementación. Elección, que se llevó a cabo durante la fase de definición de requisitos. De este modo cada miembro del grupo tendría como misión tanto la implementación como las pruebas del algoritmo implementado. Las fases de investigación, análisis, pruebas finales y redacción de documentación fueron realizadas de forma conjunta por los tres miembros del equipo. La asignación de algoritmos quedó de la siguiente manera:

Luis Lópiz Morales: Algoritmo de Agrupamiento Borroso (Fuzzy).  
Fernando Yubero Yubero: Algoritmo de Lloyd.  
Elena Fernández Albillos: Algoritmo Bayesiano.

### 2.2 PLAN DE FASE.

A continuación se detalla la planificación inicial que se pensó para llevar a cabo nuestro proyecto siguiendo el proceso unificado de desarrollo. Cada una de las fases está compuesta por una o varias iteraciones, esto es, sus actividades pueden ser repetidas varias veces, hasta completar los objetivos.

#### 2.1.1 FASE DE INICIO.

Fecha de inicio: 20 – 10 - 2008

Fecha de conclusión: 25 – 11 - 2008

Esta fase está compuesta por una iteración con los siguientes objetivos:

- Primeras reuniones con el tutor del proyecto, información de las características del mismo.
- Análisis de riesgos.
- Comienzo del diseño de los requisitos del proyecto.
- Redacción de los documentos de requisitos originales.
- Establecimiento del plan de fase del proyecto.
- Elección del entorno de desarrollo y el lenguaje de programación.

- Investigación sobre las nuevas tecnologías necesarias para la elaboración del proyecto.
- Decisión sobre la elección de los tres algoritmos a utilizar y estudio del funcionamiento de los mismos.
- Establecimiento de los objetivos para la siguiente fase.
- Elaboración de documentación.

### 2.1.2 FASE DE ELABORACIÓN.

Fecha de inicio: 25 – 11 - 2008

Fechas de conclusión: 15 – 12 - 2008

Esta fase está compuesta por una única iteración conteniendo los siguientes objetivos:

- Diseño de la arquitectura de forma que ésta sea estable, lo cual nos permitirá dividirla en módulos y así poder asignar cada uno de ellos a un miembro del grupo para que se encargue de su desarrollo en la fase de construcción.
- Construcción del primer prototipo, muy sencillo, que realice las dos fases del proyecto: entrenamiento y clasificación.
- Presentación de este primer prototipo simple al tutor del proyecto.
- Especificación de los casos de uso.
- Establecimiento de los objetivos para la siguiente fase.
- Elaboración y verificación de documentación.

### 2.1.3 FASE DE CONSTRUCCIÓN.

Compuesta por tres iteraciones, tal y como se especifica seguidamente:

#### **ITERACIÓN 1**

Fecha de inicio: 15 – 12 -2008

Fecha de conclusión: 20 – 1 – 2009

Objetivos:

- Elaboración de una primera versión del interfaz.

- Aplicando la arquitectura diseñada en la fase anterior, se comienzan a desarrollar los tres algoritmos de los que consta el proyecto, encargándose cada uno de los miembros de uno de ellos, tal y como se ha mencionado previamente.
- Estudio de nuevos requisitos.
- Elaboración de documentación.

## **ITERACIÓN 2**

Fecha de inicio: 20 – 1 - 2009

Fecha de conclusión: 25 – 2 – 2009

Objetivos:

- Desarrollo de la versión final de los algoritmos, asegurando que éstos cumplen la totalidad de los requisitos.
- Búsqueda de soluciones para los posibles riesgos presentados.
- Mejora del primer prototipo de interfaz desarrollado.
- Elaboración de un primer prototipo con todos los algoritmos integrados.
- Presentación al tutor del prototipo, y consulta de posibles mejoras.
- Revisión de riesgos detectados.
- Elaboración y verificación de la documentación.

## **ITERACIÓN 3**

Fecha de inicio: 25 – 2 - 2009

Fechas de conclusión: 31 – 3 – 2009

Objetivos:

- Elaboración del interfaz final de la aplicación
- Revisión del proyecto para la búsqueda de soluciones para los riesgos detectados hasta el momento.
- Solución de riesgos detectados e implementación final del proyecto.
- Una vez añadimos la funcionalidad de la actualización de centros después de la clasificación realizada por cada algoritmo, observamos que en el caso del clasificador Bayesiano esta actualización suponía

un coste en tiempo muy elevado, tras consultarlo con el tutor se decidió eliminar de este algoritmo la nueva funcionalidad.

- Presentación de la versión final del proyecto al tutor.
- Elaboración de la documentación y verificación de la misma.

#### 2.1.4 FASE DE TRANSICIÓN.

Cada uno de los miembros del grupo realizamos las pruebas acordadas para comprobar el correcto funcionamiento de la aplicación. Finalmente se decidió no incorporar nuevas funcionalidades al mismo. La carga de trabajo en el apartado de documentación fue superior a lo previsto en esta fase, ya que había que revisar toda la documentación anterior y realizar varias modificaciones. Se alcanzaron todos los objetivos en las fechas previstas.

- Diseño de un plan de pruebas, dichas pruebas consistirán en comprobar que funciona de la forma deseada para distintos tipos de archivos de imagen e imágenes de muy distintos tamaños según su carga en memoria.
- Posible incorporación de alguna nueva funcionalidad al proyecto.
- Revisión de la documentación realizada hasta el momento, presentación al tutor y ampliación o modificación de la misma según sus recomendaciones.
- Reunión con el tutor para la discusión de los puntos a tratar en la memoria final del proyecto.
- Elaboración de la memoria final del proyecto.
- Elaboración de la presentación final del proyecto en powerpoint.

### 3 ANÁLISIS

Este apartado recoge el estudio de los algoritmos utilizados, así como la definición de los requisitos de la aplicación y el análisis de riesgos de la misma.

#### 3.1 ESPECIFICACIÓN DEL PROYECTO.

Se describen a continuación los algoritmos elegidos para su implementación: cuantización vectorial, Algoritmo de Lloyd, Agrupamiento Borroso y clasificador Bayesiano. A partir de ellos se definen los requisitos necesarios para su implementación, los parámetros requeridos y la definición del interfaz.

##### 3.1.1 ALGORITMO DE CUANTIZACIÓN VECTORIAL.

El papel de este algoritmo en la aplicación ha sido el de llevar a cabo el entrenamiento no supervisado, lo que significa, que la aplicación no requiere la presencia de un experto que guíe el aprendizaje.

A continuación repasaremos la teoría en la que se ha basado su implementación:

Este algoritmo asume que el número de clases no se conoce inicialmente. Por tanto, se comienza suponiendo una única clase. El algoritmo progresa como sigue:

1. Para cada patrón se calcula su distancia con todos los centros existentes. Para el primer elemento él mismo constituye el primer centro.
2. Tomar el centro más cercano utilizando una medida de distancia, por ejemplo la euclídea.
3. Si dicha distancia es menor que un umbral determinado previamente, se asocia el elemento a la clase y se calcula la media de todos los elementos que pertenecen a dicha clase. Esta media nos proporciona el nuevo centro.
4. Si la distancia es mayor que el umbral prefijado se crea una nueva clase, asignando el valor del centroide al del elemento.

El sistema puede ser entrenado con cualquier umbral entero positivo, sin embargo, teniendo en cuenta el propósito para el que está pensada la aplicación y de cara a obtener un resultado satisfactorio, en función de la experiencia acumulada durante el desarrollo del proyecto, es recomendable utilizar valores comprendidos entre 80 y 150.

##### 3.1.2 ALGORITMO DE LLOYD.

Éste ha sido uno de los algoritmos utilizados por el sistema para la clasificación de las texturas de las imágenes.

Como en el punto anterior, introduciremos los fundamentos teóricos seguidos para la correcta implementación del algoritmo:

El algoritmo generalizado de Lloyd es un algoritmo de agrupamiento donde el objetivo consiste en determinar los centros  $\mathbf{c}_j$  de los  $j = 1, \dots, m$  clústeres.

1) Inicio: dados los puntos de datos  $\mathbf{x}(k)$ ,  $k = 1, 2, \dots$ , y centros de salida iniciales  $\mathbf{c}_j(0)$ ,

2) Determinar el centro  $\mathbf{c}_j(k)$  más próximo al punto  $\mathbf{x}(k)$

$$j = \arg \min_j \|\mathbf{x}(k) - \mathbf{c}_j(k)\|^2 \quad (1)$$

El hallazgo del centro más próximo se denomina competición (entre centros) dentro de la terminología utilizada en redes neuronales.

3) Actualizar el centro de salida utilizando las ecuaciones siguientes,

$$\begin{aligned} \mathbf{c}_j(k_j + 1) &= \mathbf{c}_j(k_j) - \gamma(k_j) \text{grad} \left( \|\mathbf{x}(k) - \mathbf{c}_j(k_j)\|^2 \right) \\ k_j &= k_j + 1 \end{aligned} \quad (2)$$

Obsérvese que cada centro puede tener su propia razón de aprendizaje, lo que se indica con  $k_j$  en  $\gamma(k_j)$ , siendo  $j = 1, \dots, m$ . El gradiente se calcula como,

$$\frac{\partial}{\partial \mathbf{c}_j} \|\mathbf{x} - \mathbf{c}_j\|^2 = 2(\mathbf{x} - \mathbf{c}_j) \quad (3)$$

con este gradiente, los centros de salida  $\mathbf{c}_j$  se actualizan según se indica,

$$\begin{aligned} \mathbf{c}_j(k_j + 1) &= \mathbf{c}_j(k_j) + \gamma(k_j) [\mathbf{x}(k) - \mathbf{c}_j(k_j)] \\ k_j &= k_j + 1 \end{aligned} \quad (4)$$

que se conoce comúnmente como la regla de aprendizaje competitivo.

Como se puede comprobar, para la aplicación práctica del algoritmo generalizado de Lloyd es necesario partir de unos determinados centros iniciales. En el caso concreto de este sistema, los centros iniciales utilizados son proporcionados por el entrenamiento llevado a cabo con el algoritmo de cuantización vectorial.

### Parámetros:

1. Razón de aprendizaje:

En cuanto a los valores que puede tomar, hay que considerar que la utilización de una razón excesivamente baja apenas produce actualización de los centros y no aporta nuevo aprendizaje. Por otro lado, si se usa un valor demasiado alto, el efecto es una excesiva modificación de los centros. Por tanto, un valor correcto de la razón de aprendizaje para el propósito de esta aplicación estaría comprendido entre 0.02 y 0.08.

### 3.1.3 ALGORITMO AGRUPAMIENTO BORROSO

Otra de las técnicas implementadas para la clasificación de las imágenes ha sido el agrupamiento borroso o fuzzy clustering según la terminología inglesa, tal y como se le conoce en la mayoría de los textos y literatura especializada.

Del mismo modo que en los puntos anteriores, se procede a explicar el contenido de la base teórica de dicho método:

El objetivo de la técnica de agrupamiento conocida como Agrupamiento borroso o Fuzzy Clustering consiste en dividir  $n$  objetos  $x \in X$  caracterizados por  $p$  propiedades en  $c$  “clusters” o grupos. Supongamos el conjunto de datos  $X = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^p$  un subconjunto del espacio real  $p$ -dimensional  $\mathfrak{R}^p$ . Cada  $x_k = \{x_{k_1}, x_{k_2}, \dots, x_{k_p}\} \in \mathfrak{R}^p$  se denomina vector de características,  $x_{k_j}$  es la  $j$ -ésima característica de la observación  $x_k$ . Por ejemplo, imaginemos que la intención es clasificar caracteres alfanuméricos utilizando como características los siete momentos invariantes de Hu. En este caso, cada carácter vendría identificado por los siete momentos, siendo  $p = 7$ .

Puesto que los elementos de un clúster deben ser tan similares entre sí como sea posible y a la vez deben ser tan diferentes a los elementos de otros clústeres como también sea posible, el proceso se controla por el uso de medidas de similitud basadas en distancias. Así la similitud o la diferencia entre dos puntos  $x_k$  y  $x_l$  puede interpretarse como la distancia entre esos puntos.

Una distancia entre dos objetos  $x_k$  y  $x_l$  es una función que toma valores reales  $d : X \times X \rightarrow R^+$  cumpliendo:

$$d(x_k, x_l) = d_{kl} \geq 0; \quad d_{kl} = 0 \Leftrightarrow x_k = x_l; \quad d_{kl} = d_{lk} \text{ y } d_{kl} \leq d_{kj} + d_{jl} \quad (5)$$

Cada partición del conjunto  $X = \{x_1, x_2, \dots, x_n\}$  puede enfocarse desde dos perspectivas: fuzzy y no fuzzy. Una partición no fuzzy se conoce en terminología inglesa como “crisp”. Si se desea realizar una partición del conjunto  $X$  en  $c$  clústeres tendremos  $S_i, \{i = 1, \dots, c\}$  subconjuntos. A partir de esta consideración se define lo que se conoce como grado de pertenencia  $\mu_{ik}$  de cada objeto  $x_k$  al subconjunto  $S_i$ . En el caso de conjuntos “crisp” un objeto  $x_k$  se dice que pertenece a un  $S_i$  dado y no pertenece al resto. Esto se expresa con los valores discretos  $\{0,1\}$  de la siguiente forma:  $\mu_{ik} = 1$  para indicar pertenece y

$\mu_{ik} = 0$  para expresar que no pertenece. Por el contrario, en el caso de conjuntos fuzzy se dice que un objeto puede pertenecer a diferentes subconjuntos y así se habla por ejemplo de que  $x_k$  pertenece a un conjunto  $S_i$  con grado de pertenencia  $\mu_{ik}$  y a  $S_j$  con grado de pertenencia  $\mu_{jk}$ . Como ejemplo, supongamos que se tienen tres conjuntos  $S_i$ ,  $S_j$  y  $S_h$ , en este caso podríamos decir que el objeto  $x_k$  pertenece a los conjuntos con los siguientes grados de pertenencia  $\mu_{ik} = 0.4$ ,  $\mu_{jk} = 0.5$  y  $\mu_{hk} = 0.1$ . Los valores tomados corresponden al intervalo continuo  $[0,1]$ .

Dado  $X = \{x_1, x_2, \dots, x_n\}$  y el conjunto  $V_{cn}$  de todas las matrices reales de dimensión  $c \times n$ , con  $2 \leq c < n$ . Se puede obtener una matriz representando la partición de la siguiente manera  $U = \{\mu_{ik}\} \in V_{cn}$ . Tanto en el supuesto "crisp" como en el fuzzy se deben cumplir las siguientes condiciones:

- 1)  $\mu_{ik} \in \{0,1\}$  *crisp* o  $\mu_{ik} \in [0,1]$  *fuzzy*  $1 \leq i \leq c; 1 \leq k \leq n$
- 2)  $\sum_{i=1}^c \mu_{ik} = 1 \quad 1 \leq k \leq n$
- 3)  $0 < \sum_{k=1}^n \mu_{ik} < n \quad 1 \leq i \leq c$

La localización de un clúster  $S_i$  se representa por su centro  $v_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_p}\} \in \mathfrak{R}^p$  con  $i = 1, \dots, c$ , alrededor del cual se concentran los objetos.

La definición básica de llevar a cabo el problema de la partición fuzzy para  $m > 1$  consiste en minimizar la siguiente función objetivo según la ecuación (6):

$$\min z_m(U; v) = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \|x_k - v_i\|_G^2 \quad (6)$$

donde  $G$  es una matriz de dimensión  $p \times p$  que es simétrica y definida positiva. Así se puede definir una norma general del tipo,

$$\|x_k - v_i\|_G^2 = (x_k - v_i)^t G (x_k - v_i) \quad (7)$$

Diferenciando la función objetivo para  $v_i$  (suponiendo constante  $U$ ) y  $\mu_{ik}$  (suponiendo constante  $v$ ) y aplicando la condición de que  $\sum_{i=1}^c \mu_{ik} = 1$ , se obtiene,

$$v_i = \frac{1}{\sum_{k=1}^n (\mu_{ik})^m} \sum_{k=1}^n (\mu_{ik})^m x_k \quad i = 1, \dots, c \quad (8)$$

$$\mu_{ik} = \frac{\left( \frac{1}{\|x_k - v_i\|_G^2} \right)^{2/m-1}}{\sum_{j=1}^c \left( \frac{1}{\|x_k - v_j\|_G^2} \right)^{2/m-1}} \quad i = 1, \dots, c; k = 1, \dots, n \quad (9)$$

El exponente  $m$  se conoce como peso exponencial y disminuye la influencia del ruido al obtener los centros de los clústeres, reduciendo la influencia de los valores pequeños de  $\mu_{ik}$  (puntos lejos de  $v_i$ ) frente a valores altos de  $\mu_{ik}$  (puntos cercanos a  $v_i$ ). Cuanto mayor sea  $m > 1$  mayor es dicha influencia.

En el algoritmo conocido como  $c$ -medias fuzzy para obtener los valores de las expresiones (4) y (5) es el siguiente:

1) Elegir  $c$  ( $2 \leq c \leq n$ ),  $m$  ( $1 < m < \infty$ ) y la matriz  $G$  de dimensión  $p \times p$  siendo simétrica y definida positiva. Inicializar  $U^{(0)}$  y poner  $t = 0$ .

2) Calcular los  $c$  centros fuzzy de los clústeres a partir de (4)  $\{v_i^{(t)}\}$  utilizando  $U^{(t)}$ .

3) Calcular los nuevos grados de pertenencia de la matriz  $U^{(t+1)}$  utilizando  $\{v_i^{(t)}\}$  a partir de la condición (5) si  $x_k \neq v_i^{(t)}$ . De lo contrario

$$\mu_{jk} = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \quad (10)$$

4) Elegir una norma matricial y calcular  $\Delta = \|U^{(t+1)} - U^t\|_G$ . Si  $\Delta > \varepsilon$  poner  $t = t + 1$  y regresar al paso 2), de lo contrario detener el proceso.

Por tanto, para llevar a cabo un proceso de agrupamiento es necesario definir: a) número de clústeres  $c$ ; b) el peso exponencial  $m$  (siendo un valor típico de 2); c) la matriz  $G$  que induce la norma; d) el método para inicializar  $U^{(0)}$  y e) el criterio de terminación  $\Delta \leq \varepsilon$ .

Especial mención requiere el estudio de la matriz  $G$  que determina la forma del clúster. Si se elige la norma Euclídea, entonces  $G$  es la matriz identidad  $I$  y la forma de los clústeres se asume que constituyen una hiper-esfera.  $G$  también se puede elegir como una matriz diagonal con  $G_D = [diag(\sigma_j^2)]^{-1}$  o la norma de Mahalanobis con  $G_M = [cov(x)]^{-1}$ , donde  $\sigma_j^2$  denota la varianza de la característica  $j$  y  $cov$  la covarianza.

Para definir el criterio de terminación se ha forzado al programa a no hacer más de cinco iteraciones, y se ha asignado a  $\Delta$  el valor de la distancia entre los

antiguos centros y los nuevos centros calculados, esto es, la suma de las distancias dividido entre el número de centros. Cuando este valor sea menor que el umbral, o se hayan completado las cinco iteraciones, el algoritmo finalizará

### **Parámetros:**

Los parámetros necesarios para el algoritmo de Agrupamiento Borroso son los que se muestran a continuación:

1. Peso exponencial  $m$ :

Toma el valor 2.

2. Número máximo de iteraciones:

Sirve para definir el criterio de terminación. El número máximo de iteraciones de nuestro algoritmo queda establecido en 5.

3. Centros iniciales  $c$ :

Se obtienen inicialmente a partir de los centros del entrenamiento.

4. Umbral Fuzzy:

Como se ha comentado antes, cuando la diferencia entre los antiguos centros y los nuevos, sea menor que el umbral terminamos la clasificación.

### 3.1.4 ALGORITMO BAYESIANO

El método basado en la teoría de la decisión de Bayes es un método estadístico clásico en clasificación de patrones. Se fundamenta en el supuesto de que el problema de la decisión se enfoca en términos probabilísticos y que todas las probabilidades relevantes resultan conocidas.

El entrenamiento para este algoritmo consistirá en calcular medias y matrices de covarianza de cada clase de las muestras proporcionadas por el entrenamiento C-Medias Fuzzy.

Por tanto la clasificación de cada píxel de una imagen se enfoca según la probabilidad de pertenecer a cada una de ellas. Esa probabilidad vendrá dada en nuestro caso por la distancia de Mahalanobis que a su vez se calcula con los datos obtenidos en el entrenamiento.

El algoritmo de clasificación Bayesiana en el ámbito de la clasificación de imágenes nos será de utilidad para analizar una imagen píxel a píxel y determina a qué clase pertenece cada uno según una probabilidad calculada.

Supongamos que no tenemos conocimiento a cerca de los valores R, G, y B del píxel que estamos analizando. Utilizando la terminología de la teoría de la

decisión, podemos decir que la probabilidad a priori de que un píxel pertenezca a cualquiera de las clases es la misma y cuya suma es la unidad.

Sea  $y$  el estado que designa la pertenencia de un píxel a una de las clases,  $c$  el número de clases en que hay que dividir la imagen y considerando a  $y$  una variable aleatoria, ya que la pertenencia a uno de los dos estados es impredecible, podemos decir de forma más precisa que suponemos que existe alguna *probabilidad a priori*  $P(y=c_i)$  de que el píxel pertenezca a la clase  $c_i$  y alguna *probabilidad a priori* de que pertenezca a la clase  $c_j$   $P(y=c_j)$  y así con todas las clases que queramos hasta  $c$ .

A la hora de decidir a cerca de la pertenencia de un píxel a una clase concreta, la única información son esas probabilidades a priori y parece razonable utilizar la siguiente *regla de decisión*: decidir la que tenga mayor probabilidad.

Pero en nuestro caso conocemos los valores para R, G y B del píxel a clasificar. A estos tres valores los llamaremos  $\mathbf{x} = \{x_R, x_G, x_B\}$ . Así, diferentes píxeles darán diferentes valores de  $\mathbf{x}$ , resulta natural por tanto expresar esta variabilidad en términos probabilísticos. En este sentido, consideremos a  $\mathbf{x}$  una variable aleatoria continua cuya distribución depende de la clase.

Sean  $p(\mathbf{x} / y = c_j)$  las funciones de *densidad de probabilidad condicionales* para  $\mathbf{x}$  dado que el píxel pertenezca a la clase  $c_i$ . Supongamos que conocemos tanto las probabilidades a priori como estas últimas funciones de densidad de probabilidad. Suponer además que medimos las componentes R, G y B de un píxel y descubrimos que vale  $x$ , la pregunta será ¿cómo influye esta medida sobre nuestra actitud con relación a la clase de que se trata? La respuesta nos la proporciona la *regla de Bayes*, considerando que tanto las probabilidades a priori  $P(y = c_j)$  como las densidades condicionales para cada clase  $p(\mathbf{x} / y = c_j)$  son conocidas o se pueden estimar, es posible determinar para una observación dada  $\mathbf{x}$  la probabilidad de que esa observación pertenezca a una determinada clase. Estas probabilidades, llamadas *probabilidades a posteriori* pueden usarse para construir una regla discriminante:

$$p(y = c_j / \mathbf{x}) = \frac{p(\mathbf{x} / y = c_j)P(y = c_j)}{p(\mathbf{x})} \quad (11)$$

donde

$$p(\mathbf{x}) = \sum_{j=1}^c p(\mathbf{x} / y = c_j)P(y = c_j) \quad (12)$$

La regla de Bayes muestra cómo la observación del valor  $\mathbf{x}$  cambia las probabilidades a priori a las *probabilidades a posteriori*  $p(y = c_j / \mathbf{x})$ .

Una vez que se determinan esas probabilidades a posteriori, la siguiente regla de decisión se utiliza para clasificar  $x$ .

$$\mathbf{x} \in c_i \text{ si } P(y = c_i / \mathbf{x}) > P(y = c_j / \mathbf{x}) \quad \forall i \neq j, \quad i, j = 1, 2, \dots, c \quad (13)$$

Ahora bien, si nos fijamos en el segundo término de la expresión que obtiene las *probabilidades a posteriori* del teorema de Bayes y eliminado el término no discriminante  $p(\mathbf{x})$  (no aporta nada en la decisión), se tiene una forma alternativa de clasificar el vector de atributos  $\mathbf{x}$ :

$$\mathbf{x} \in c_i \text{ s\u00ed } P(\mathbf{x}/y=c_i)P(y=c_i) > P(\mathbf{x}/y=c_j)P(y=c_j) \quad \forall i \neq j, i, j = 1, 2, \dots, c \quad (14)$$

Generalmente las distribuciones de densidad de probabilidad se eligen Normales o Gaussianas. Un caso especial surge cuando las probabilidades a priori son iguales para todas las clases, ya que en esta situaci\u00f3n la distancia de Mahalanobis se puede utilizar como funci\u00f3n discriminante mediante la siguiente regla de decisi\u00f3n a partir de la regla anterior y teniendo en cuenta el signo negativo en el t\u00e9rmino exponencial de la funci\u00f3n de densidad de probabilidad Normal, as\u00ed

$$\mathbf{x} \in c_i \text{ s\u00ed } d_M^2(\mathbf{x}, \mathbf{m}_i) < d_M^2(\mathbf{x}, \mathbf{m}_j) \quad \forall i \neq j, i, j = 1, 2, \dots, c \quad (15)$$

donde  $\mathbf{m}_i, \mathbf{m}_j$  son los vectores media de las clases  $c_i$  y  $c_j$  respectivamente.

Sin p\u00e9rdida de generalidad, la distancia de un vector  $\mathbf{x}_k$  a la clase  $c_i$  resulta ser:

$$d_M^2(\mathbf{x}_k, \mathbf{m}_i) = (\mathbf{x}_k - \mathbf{m}_i)' C_i^{-1} (\mathbf{x}_k - \mathbf{m}_i) \quad (16)$$

En el supuesto de que las matrices de covarianza sean la identidad, la distancia de Mahalanobis al cuadrado resulta ser la distancia Eucl\u00eddea al cuadrado, en cuyo caso tendr\u00edamos,

$$d_E^2(\mathbf{x}, \mathbf{m}_i) = (\mathbf{x} - \mathbf{m}_i)' (\mathbf{x} - \mathbf{m}_i) = \mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{m}_i + \mathbf{m}_i'\mathbf{m}_i \quad (17)$$

En la expresi\u00f3n anterior el t\u00e9rmino  $\mathbf{x}'\mathbf{x}$  no discrimina, ya que se repite en todas las clases, de forma que puede despreciarse. Ahora, si se cambia de signo y se divide por 2 en la ecuaci\u00f3n anterior se obtiene la siguiente funci\u00f3n discriminante, donde se deduce que la distancia Eucl\u00eddea al cuadrado m\u00ednima hace la expresi\u00f3n siguiente m\u00e1xima:

$$fd_i(\mathbf{x}) = \mathbf{x}'\mathbf{m}_i - \frac{1}{2}\mathbf{m}_i'\mathbf{m}_i \quad (18)$$

## Entrenamiento

El entrenamiento, como paso previo a la clasificaci\u00f3n Bayesiana consistir\u00e1 en obtener las medias y matrices de covarianza de las muestras.

Una muestra consiste en conocer un p\u00edxel y la clase a la que pertenece.

En este caso las muestras nos las proporcionar\u00e1 el algoritmo de entrenamiento C-Medias Fuzzy, que normalmente nos dar\u00e1 la informaci\u00f3n necesaria, suficiente y fiable a partir de una imagen de entrenamiento, dependiendo de los par\u00e1metros escogidos para dicho algoritmo.

En el caso de que se quisiera clasificar varias imágenes también es posible obtener más muestras una vez hayamos finalizado la clasificación de una imagen, añadiendo a la base de muestras la salida obtenida y refinando por tanto la calidad de las muestras para utilizarlas en posteriores clasificaciones.

Una vez obtenidas las muestras se procede a calcular para cada clase la media y la covarianza como se muestra a continuación:

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad C = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^t$$

El valor de las medias y covarianzas obtenidas serán posteriormente utilizadas para realizar la clasificación.

### **Parámetros:**

Los parámetros necesarios para el algoritmo Bayesiano son los que se muestran a continuación:

1. Número de clases 'c'.

Se utiliza en el entrenamiento y en la clasificación. Este parámetro indica el número de clases en que se desea clasificar los píxeles de una imagen.

2. Muestras clasificadas

Este parámetro se corresponde con las muestras del entrenamiento, sobre las que se calculan medias y matrices de covarianzas.

3. Medias ' $\mathbf{m}_i$ ' y matrices de Covarianza ' $\mathbf{C}_i$ '

Si se han guardado resultado de previas clasificaciones puede ser de gran utilidad utilizar las medias y matrices de covarianzas obtenidas. Los subíndices  $i$  se refieren a cada una de las clases.

4. Parámetros del algoritmo C-Medias Fuzzy.

Puesto que para obtener las muestras iniciales clasificadas para nuestro entrenamiento utilizamos el algoritmo C-Medias Fuzzy, serán necesarios unos valores apropiados para su aplicación.

## 3.2 REQUISITOS

Una vez que se han definido los métodos, su funcionamiento y los parámetros requeridos por cada uno de los métodos anteriormente expuestos, se procede a la definición de los requisitos funcionales, que ha de cumplir la aplicación para su correcto funcionamiento. Dentro de tales requisitos se especifican aquellos relacionados con la interfaz de usuario.

### 3.2.1 REQUISITOS FUNCIONALES

- El sistema será capaz de procesar los siguientes formatos de imagen: bmp, jpeg, y png.
- La aplicación permitirá el procesado de imágenes de cualquier tamaño.
- El programa debe incluir la capacidad de entrenamiento no supervisado, esto es, clasificar los píxeles de una imagen en un determinado número de clases de forma automática, sin la ayuda de un experto.
- Existirá la posibilidad de clasificar los píxeles de una imagen mediante tres algoritmos diferentes: algoritmo de Bayes, Agrupamiento borroso y algoritmo de Lloyd.
- La clasificación de una imagen actualizará el valor de los parámetros aprendidos, generalmente los centros de las clases. .
- El sistema permitirá al usuario la configuración de los parámetros de ejecución desde la interfaz.
- El programa ofrecerá los resultados del entrenamiento en modo texto (con los valores de las clases) y los de la clasificación en formato imagen y texto (con los valores de las clases igualmente).
- El usuario podrá elegir las imágenes utilizadas en la clasificación o en el entrenamiento mediante un menú de fácil uso y comprensión.
- La aplicación ofrecerá al usuario la posibilidad de guardar en disco la imagen una vez haya sido clasificada.
- El sistema guardará los resultados de los entrenamientos en disco automáticamente.

### 3.2.2 REQUISITOS NO FUNCIONALES.

- Para el correcto funcionamiento del sistema, el equipo deberá contar con una memoria mínima de 512Gb, una frecuencia de procesados de 1,5Ghz y un espacio en disco de 50MB.
- Los tiempos de entrenamiento y clasificación no deberán superar los 60 minutos en ambos casos.

## 3.3 RIESGOS

A continuación se procede al análisis de riesgos. Se tendrán en cuenta las siguientes categorías de riesgos: tecnológicos, de personal y de organización.

### 3.3.1 TECNOLÓGICOS.

Podemos definir riesgo tecnológico como la posibilidad de que existan consecuencias indeseables o inconvenientes de un acontecimiento relacionado con el acceso o uso de la tecnología y cuya aparición no se puede determinar a priori.

Información Riesgo Tecnológico 1			
Probabilidad	Alta	Consecuencias	Serias
Tipo	Predecible	Estado Actual	Presente
Descripción	Desconocimiento de la tecnología a utilizar.		
Plan de contingencia	Utilización de tutoriales y documentación de la tecnología.		

Información Riesgo Tecnológico 2			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Limitaciones en las funcionalidades de la librería Java Advanced Image.		
Plan de contingencia	Utilización de una librería alternativa.		

Información Riesgo Tecnológico 3			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Falta de memoria en la Máquina Virtual de java		
Plan de contingencia	Disponibilidad de ordenadores suficientemente potentes.		

Información Riesgo Tecnológico 4			
Probabilidad	Media	Consecuencias	Moderadas
Tipo	Posible	Estado Actual	No presente

Descripción	Tiempo de ejecución demasiado elevado.
Plan de contingencia	Diseño optimizado del código.

Información Riesgo Tecnológico 5			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Pérdida de código, resultados e información por un fallo tecnológico.		
Plan de contingencia	Guardar varias copias de todo el material.		

Información Riesgo Tecnológico 6			
Probabilidad	Baja	Consecuencias	Moderadas
Tipo	Impredecible	Estado Actual	No presente
Descripción	Utilización de versiones no actualizadas por parte de los miembros del grupo		
Plan de contingencia	Uso de un repositorio de código		

### 3.3.2 DE PERSONAL

Podemos definir riesgos de personal como aquellos cambios en las condiciones personales de los integrantes del equipo que afectan o pueden afectar al desarrollo del proyecto.

Información Riesgo de Personal 1			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Baja de alguno de los miembros del grupo.		

Plan de contingencia	Conocimiento total de todos los integrantes del grupo de las tecnologías utilizadas y el estado de la aplicación en todo momento, para poder redistribuir el trabajo pendiente.
----------------------	---

### 3.3.3 ORGANIZATIVOS.

Calificamos como riesgos organizativos o de organización todos aquellos relativos a las relaciones interpersonales dentro del equipo que afectan al funcionamiento del grupo.

Información Riesgo Organizativo 1			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Falta de comunicación entre los integrantes del grupo.		
Plan de contingencia	Reuniones periódicas y contacto vía e-mail.		

Información Riesgo Organizativo 2			
Probabilidad	Baja	Consecuencias	Serias
Tipo	Impredecible	Estado Actual	No presente
Descripción	Discrepancia de ideas dentro del grupo de trabajo.		
Plan de contingencia	Tolerancia y disposición de escucha por parte de todos los miembros.		



## 4 DESARROLLO

En este apartado se detalla el proceso de desarrollo seguido y el alcance de objetivos con respecto a la planificación planteada en el apartado dos.

### 4.1 FASES DE DESARROLLO

#### 4.1.1 FASE DE INICIO.

Después de las primeras reuniones con el tutor, quedaron establecidos los requisitos originales del proyecto. En las fechas planificadas se cumplieron por tanto los siguientes objetivos:

- Captura de requisitos.
- Análisis de riesgos.
- Establecimiento del plan de fase inicial del proyecto.
- Elección del tipo de entorno de desarrollo y el lenguaje de programación. Decidimos utilizar Java como lenguaje de programación y Eclipse como entorno de desarrollo ya que todos los miembros del grupo habíamos trabajado varias veces en este entorno y estábamos familiarizados con el mismo
- Investigación sobre las nuevas tecnologías necesarias para la elaboración del proyecto. Tras las primeras reuniones con el profesor seguimos su recomendación sobre la utilización de la biblioteca JAI (Java Advanced Imaging) para el tratamiento de imágenes. Así que en esta fase se produjo el aprendizaje y familiarización de dicha biblioteca, ya que nos facilitaba notablemente el trabajo con los píxeles de la imagen.
- Decisión de los tres algoritmos que vamos a utilizar e investigación de los mismos. Se decidió entre los tres miembros del grupo que los algoritmos a implementar fueran: Lloyd, Agrupamiento borroso y Bayessiano. Comenzamos su investigación con las recomendaciones bibliográficas del tutor.
- Establecimiento de los objetivos para la siguiente fase.
- Elaboración de documentación.

#### 4.1.2 FASE DE ELABORACIÓN.

El objetivo más importante de los alcanzados en esta fase fue el correspondiente al diseño de la arquitectura de la aplicación; aunque en esta fase no quedó establecido el diseño definitivo, ya que dejamos abierta la posibilidad de añadir nuevas clases en la siguiente fase. El diseño inicial del que

disponíamos nos permitió dividir nuestra aplicación en módulos diferenciados, lo cual facilitó la división del trabajo a realizar por cada miembro del grupo.

Cabe destacar que, como paso previo al prototipo planteado como objetivo en la planificación, que tenía que realizar entrenamiento y clasificación con un sólo algoritmo (cuantización vectorial), se realizó otro prototipo muy sencillo, que nos permitía la modificación del color de los píxeles de la imagen que se le pasaba como entrada, en este caso utilizando las funcionalidades de la biblioteca JAI.

El resto de objetivos planteados en esta fase se alcanzaron sin problemas ni dificultades destacables:

- Presentación de este primer prototipo simple al tutor del proyecto.
- Especificación de los casos de uso.
- Establecimiento de los objetivos para la siguiente fase.
- Elaboración de documentación.

#### 4.1.3 FASE DE CONSTRUCCIÓN.

Siguiendo el esquema de iteraciones expuesto previamente, a continuación se especifica el grado de cumplimiento de los diferentes objetivos involucrados.

##### **ITERACIÓN 1**

Se cumplieron la totalidad de los objetivos marcados en las fechas previstas:

- Elaboración de una primera versión del interfaz.
- Aplicando la arquitectura diseñada en la fase anterior, se comienza a desarrollar los tres algoritmos de los que consta el proyecto, encargándose cada uno de los miembros de uno de ellos.
- Como caso particular y digno de mención, en el desarrollo del algoritmo Bayesiano nos dimos cuenta de la necesidad del uso de una biblioteca que facilitara el trabajo con matrices y las operaciones con las mismas (sobre todo la inversa) tras consultar con el tutor nos decidimos por la biblioteca JAMA.
- Estudio de nuevos requisitos.
- Elaboración de documentación.

## **ITERACIÓN 2**

En esta iteración, igualmente, los objetivos siguientes se alcanzaron según la planificación temporal prevista:

- Búsqueda de soluciones para los posibles riesgos presentados.
- Elaboración de un primer prototipo con todos los algoritmos integrados.
- Presentación al tutor del prototipo, y consulta de posibles mejoras.
- Revisión de riesgos detectados.
- Elaboración de documentación.

No obstante, los dos siguientes objetivos no se alcanzaron en su totalidad:

- Desarrollo de la versión final de los algoritmos, asegurando que éstos cumplieran la totalidad de los requisitos.
- Se desarrolló la mayor parte de la funcionalidad prevista para los algoritmos seleccionados, con la salvedad de que éstos no realizaban un entrenamiento del sistema (actualización de los centros) una vez que terminaban la clasificación de la imagen. La implementación de esta funcionalidad se dejó para la siguiente iteración
- Mejora del primer prototipo de interfaz presentado. Este objetivo no se consiguió en esta iteración ya que se continuó utilizando la versión anterior del interfaz, centrándonos así en la implementación de la versión final de los algoritmos.

## **ITERACIÓN 3**

Los objetivos alcanzados en esta tercera iteración fueron:

- Elaboración del interfaz final de la aplicación
- Revisión del proyecto para la búsqueda de soluciones frente los riesgos detectados.
- Solución de riesgos detectados e implementación final de la aplicación.
- Presentación de la versión final de la aplicación al tutor.
- Elaboración de documentación

#### 4.1.4 FASE DE TRANSICIÓN.

Cada uno de los miembros del grupo realizamos las pruebas acordadas para comprobar el correcto funcionamiento de la aplicación. Finalmente se decidió incorporar una nueva opción en el interfaz que permitiera recuperar los resultados obtenidos durante una clasificación de datos, que fueron previamente almacenados. Esto permite evitar la necesidad de tener que pasar así por la fase de entrenamiento.

La carga de trabajo en el apartado de documentación fue superior a lo previsto en esta fase ya que había que revisar toda la documentación anterior y realizar varias modificaciones. No obstante se alcanzaron todos los objetivos en las fechas previstas.

- Diseño de un plan de pruebas. Estas pruebas consistían en comprobar el correcto comportamiento de la aplicación para distintos tipos de archivos de imagen e imágenes de muy distintos tamaños para su carga en la memoria del computador.
- Posible incorporación de alguna nueva funcionalidad al proyecto, tal como la ya mencionada relativa a la mejora del interfaz.
- Revisión de la documentación elaborada hasta el momento, presentación al tutor y ampliación o modificación de la misma según sus recomendaciones.
- Reunión con el tutor para la discusión de los puntos a tratar en la memoria final del proyecto.
- Elaboración de la memoria final del proyecto.
- Elaboración de la presentación final del proyecto en PowerPoint.

#### 4.2 PLAN DE PRUEBAS.

Consta de varias iteraciones, para cada una de ellas se establecieron el tipo de pruebas a realizar, sus objetivos y el código necesario para poder llevarlas a cabo.

##### Primera iteración de pruebas:

La primera iteración de pruebas se corresponde con las realizadas para comprobar el correcto funcionamiento del algoritmo de clasificación no supervisado implementado, concretamente el ya comentado Cuantización Vectorial.

##### **Pruebas definidas:**

- Verificación de la correcta clasificación en clases de los píxeles individuales correspondientes a la imagen de entrada, proporcionada como entrada al sistema.

- Verificación y validación del correcto funcionamiento de la variable umbral, se debe obtener para una misma imagen diferente número de clases según se varíe el valor de dicho umbral.
- Verificación y validación del correcto funcionamiento de la aplicación independientemente del tipo de imagen suministrada como entrada. Se realizaron pruebas tanto con imágenes del tipo “.bmp”, como “.gif”, o “.jpg”.

### **Resultados de las pruebas:**

- La clasificación en clases realizada por el algoritmo de cuantización vectorial es la correcta y por tanto coincide con lo esperado.
- El umbral determina diferente número de clases, tal y como estaba previsto.
- El algoritmo acepta diferentes formatos de imágenes.
- Se observó que el tiempo de ejecución era demasiado elevado para imágenes a partir de tamaños de 500x500 píxeles.

### Segunda iteración de pruebas:

La segunda iteración de pruebas fue realizada de forma independiente por cada uno de los componentes del grupo, encargándose cada uno de ellos de comprobar el correcto funcionamiento del algoritmo implementado, es decir de los tres algoritmos previstos: Agrupamiento borroso, Lloyd o Bayesian.

Paralelamente se prueba la carga de imágenes mediante el interfaz, así como la presentación de las mismas sobre las ventanas proporcionadas por el propio interfaz.

### **Pruebas:**

- El algoritmo de Lloyd realiza una correcta clasificación de los píxeles de la imagen.
- El algoritmo basado como agrupamiento borroso realiza igualmente una correcta clasificación de los píxeles de la imagen proporcionada.
- El algoritmo Bayesiano, al igual que los anteriores, realiza una correcta clasificación de los píxeles de la imagen.
- El interfaz carga y muestra las imágenes de forma correcta.

### **Resultados de las pruebas:**

- Los algoritmos Lloyd, Agrupamiento Borroso y Bayesiano trabajan y funcionan correctamente, según lo esperado.

- La carga y presentación de imágenes en el interfaz es correcta.

Durante el proceso de evaluación de ésta iteración de pruebas, se decidió redimensionar la imagen para su correcta presentación en el interfaz siempre que sus dimensiones fueran superiores a 500x500 píxeles. No obstante, esta consideración sólo se realiza durante la presentación de la imagen, no así en lo que respecta al proceso de clasificación, que se realiza sobre el contenido total de la imagen original.

#### Tercera iteración de pruebas:

La tercera iteración de pruebas se corresponde con las pruebas de integración de los tres algoritmos y el interfaz. Estas pruebas fueron realizadas por los tres miembros del grupo, centrándose cada uno de ellos en la comprobación del funcionamiento de los algoritmos de cuya implementación no se había encargado.

También se comprueba que el interfaz muestra de forma correcta la imagen una vez clasificada. Si bien en la iteración anterior sólo se comprueba que cada algoritmo genere una correcta clasificación en clases de los píxeles de la imagen, en esta iteración se comprueba que el interfaz sea capaz de presentar esa imagen una vez asignado a cada píxel un color diferente según la clase a la que pertenece.

#### **Pruebas:**

- Los tres algoritmos siguen funcionando de la forma esperada una vez se ha producido su integración en el interfaz.
- Los algoritmos devuelven los píxeles de la imagen clasificados para que ésta pueda ser mostrada de forma correcta en el interfaz.
- La asignación de colores para representar cada clase es adecuada independientemente del número de clases.

#### **Resultados de las pruebas:**

- Los algoritmos integrados en el interfaz funcionan de forma correcta.
- Los algoritmos devuelven los píxeles clasificados.
- El interfaz presenta de forma correcta esta clasificación, asignando una etiqueta distinta en forma de color según la clase de pertenencia.

Durante la evaluación de esta iteración de pruebas se decidió ampliar el número de etiquetas de color para la imagen una vez clasificada. Esto se justifica por el hecho de que si se elige un umbral con valor bajo para la determinación del número de clases por el algoritmo no supervisado de cuantización vectorial, el número de clases obtenido durante el entrenamiento es elevado, por lo que es necesario disponer de un número suficiente de etiquetas de clase.

### Cuarta iteración de pruebas:

Esta última iteración de pruebas también fue realizada por los tres miembros del grupo. Se centra particularmente en comprobar la última versión del interfaz, con las mejoras introducidas al mismo. También se comprueba que después de la clasificación realizada por cada algoritmo, se actualicen de forma correcta las clases, con los nuevos datos obtenidos a partir de dicha clasificación.

#### **Pruebas:**

- El interfaz es capaz de pasar a cada algoritmo los parámetros necesarios para su funcionamiento, estos parámetros son especificados por el usuario.
- Tras la ejecución de los algoritmos, el fichero que contiene los valores de las distintas clases, se actualiza con los nuevos datos procedentes de la ejecución de los algoritmos, para utilizar estos datos en posteriores clasificaciones.
- Se puede cargar un fichero con una clasificación de clases inicial desde el interfaz (fichero “umbral”), si el usuario lo desea, y por el contrario no desea pasar por la fase de entrenamiento.

#### **Resultados de las pruebas:**

- Los parámetros introducidos a través del interfaz por el usuario, se pasan de forma correcta a los algoritmos, el funcionamiento de cada algoritmo continúa siendo el esperado.
- Después de la ejecución de los algoritmos se produce la actualización del fichero de clases. En el caso del método de clasificación Bayesiano se observa un coste de tiempo elevado al realizar dicha actualización.
- La carga del fichero denominado “umbral” (archivo “.bin” que contiene los valores RGB tanto de los centros de las clases, como de píxeles que componen cada clase) con los valores iniciales de las clases es correcta.

Durante el proceso de evaluación de esta iteración de pruebas se decidió eliminar la funcionalidad de actualizar las clases tras la clasificación en el caso del algoritmo Bayesiano debido al elevado coste computacional observado.



## 5 DISEÑO

En este apartado recogemos en primer lugar los aspectos más importantes del diseño de la aplicación, indicando como hemos repartido la funcionalidad del sistema en paquetes. A continuación se detallan los aspectos más importantes de la arquitectura del sistema a través de los diagramas UML y por último mostramos la interacción entre objetos en dos casos de uso de nuestro sistema a través de los diagramas de secuencia.

### 5.1 ESPECIFICACIÓN DEL DISEÑO.

Como aspecto más relevante del diseño de la aplicación se puede destacar una estructura de paquetes que proporciona la modularidad deseada. Cada paquete engloba clases cuya funcionalidad se restringe a un área bien diferenciada.

Los paquetes de los que consta el sistema son: Interfaz, UtilidadesImagen y Algoritmos.

El paquete Interfaz controla todo el funcionamiento de la aplicación desde la interfaz de usuario. Por su parte, el paquete UtilidadesImagen se encarga del preprocesado de las imágenes y del tratamiento de los archivos. Por último, es el paquete Algoritmos el que contiene toda la lógica del sistema, que se concreta básicamente en entrenamiento y clasificación.

Dentro del paquete Algoritmos, se encuentra una superclase Algoritmo de la que heredan todos los algoritmos del sistema, tanto el algoritmo de cuantización vectorial (utilizado en el entrenamiento) como los algoritmos de Bayes, de Agrupamiento Borroso y Lloyd (clasificación de la imagen).

El esquema de la organización en paquetes de la aplicación propuesta se muestra en la figura 1.

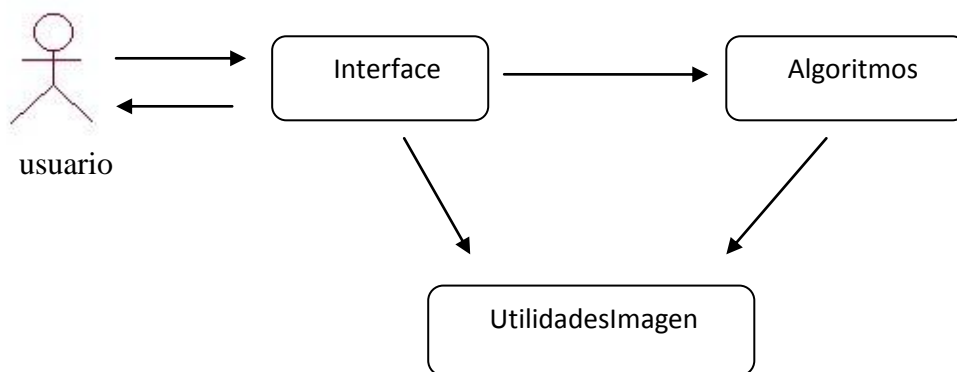


Figura 1. Esquema de paquetes de la aplicación

El agente que interactúa con nuestra aplicación es únicamente el usuario, que a través del interface selecciona la imagen a clasificar y el algoritmo con el que se hará dicha clasificación. Introduce todos los parámetros requeridos por el algoritmo a utilizar, tras lo cual devuelve los resultados, mostrando en los casos previstos, los resultados por

pantalla, principalmente los que aparecen en forma de imagen relativos a las clasificaciones de las texturas.

## 5.2 DIAGRAMAS UML.

En este punto se presentan los diagramas de clases de cada paquete:

### Interface:

El diagrama correspondiente al paquete Interface se muestra en la Figura 2. Se muestra la relación de dependencia de la clase Interfaz a las clases Algoritmo (paquete Algoritmos) y TratamientoFichero (paquete UtilidadesImagen).

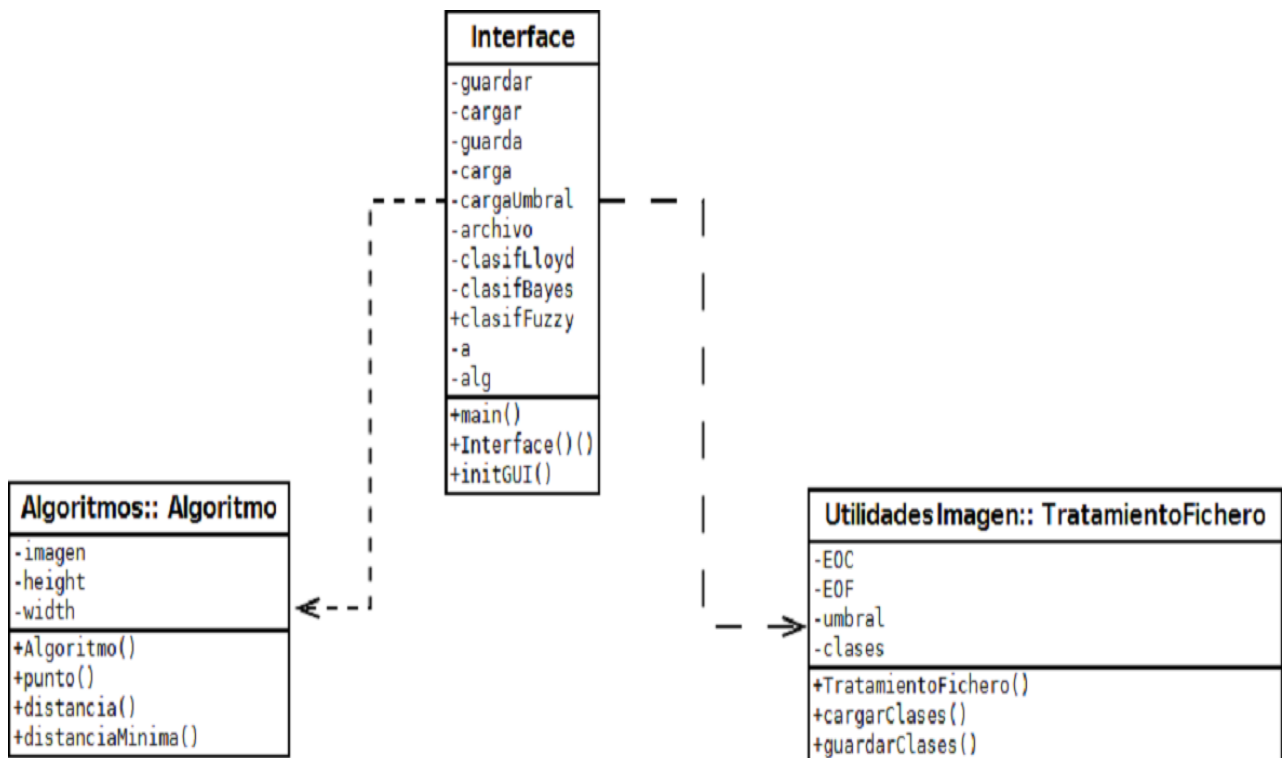


Figura 2. Diagrama de clases del paquete Interface

### UtilidadesImagen:

La Figura 3 muestra el diagrama de clases del paquete UtilidadesImagen. En el diagrama se observa las relaciones de dependencia entre las clases TratamientoImagen, Clase, colores y TratamientoFichero y la clase Pixel.

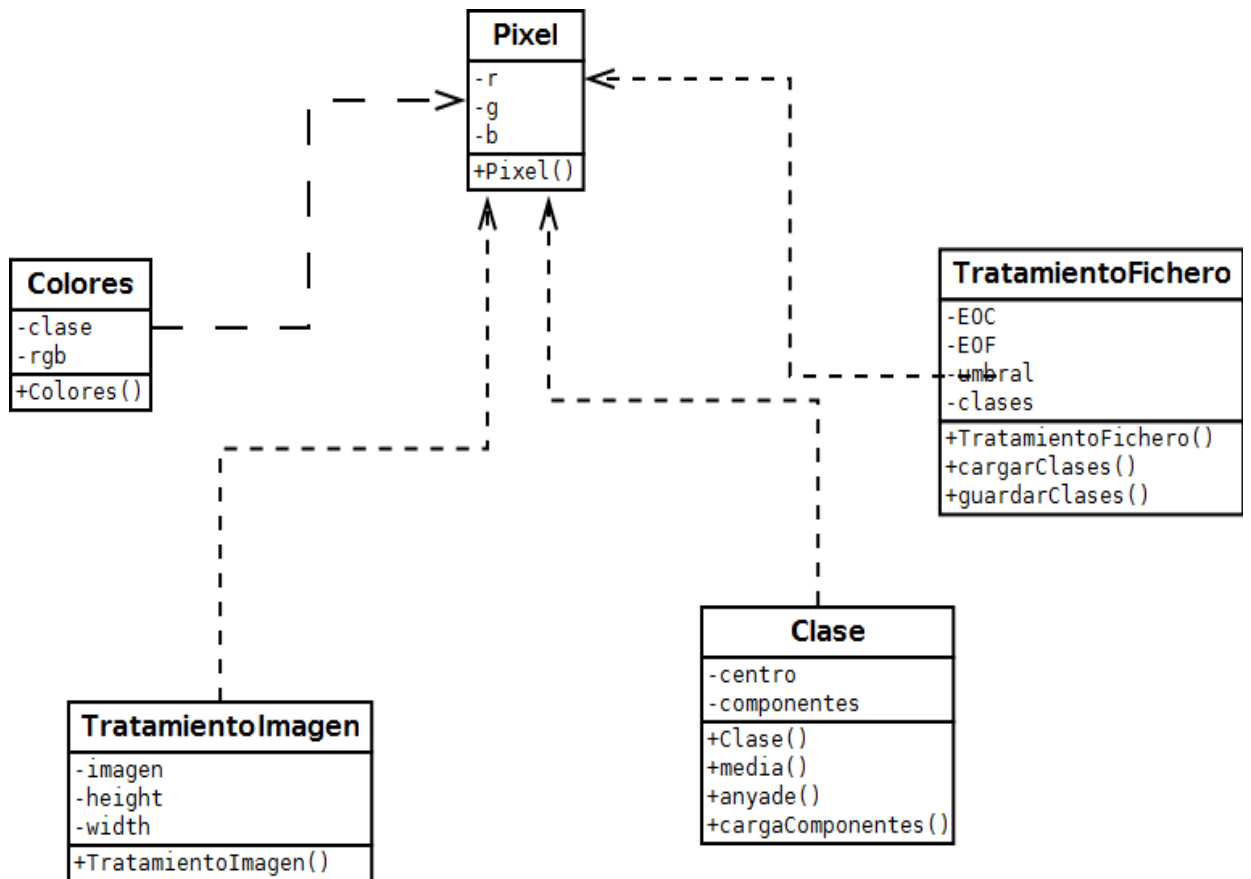


Figura 3. Diagrama de clases del paquete UtilidadesImagen

Algoritmos:

La Figura 4 refleja el diagrama de clases del paquete Algoritmos.

En él se puede observar las relaciones de dependencia de las clases Fuzzy (Agrupamiento Borroso), Lloyd y Bayesiano (paquete Algoritmos) hacia las clases Pixel y Clase (UtilidadesImagen). Además, destaca la relación de herencia de las clases Fuzzy, Lloyd y Bayesiano hacia la clase Algoritmo.

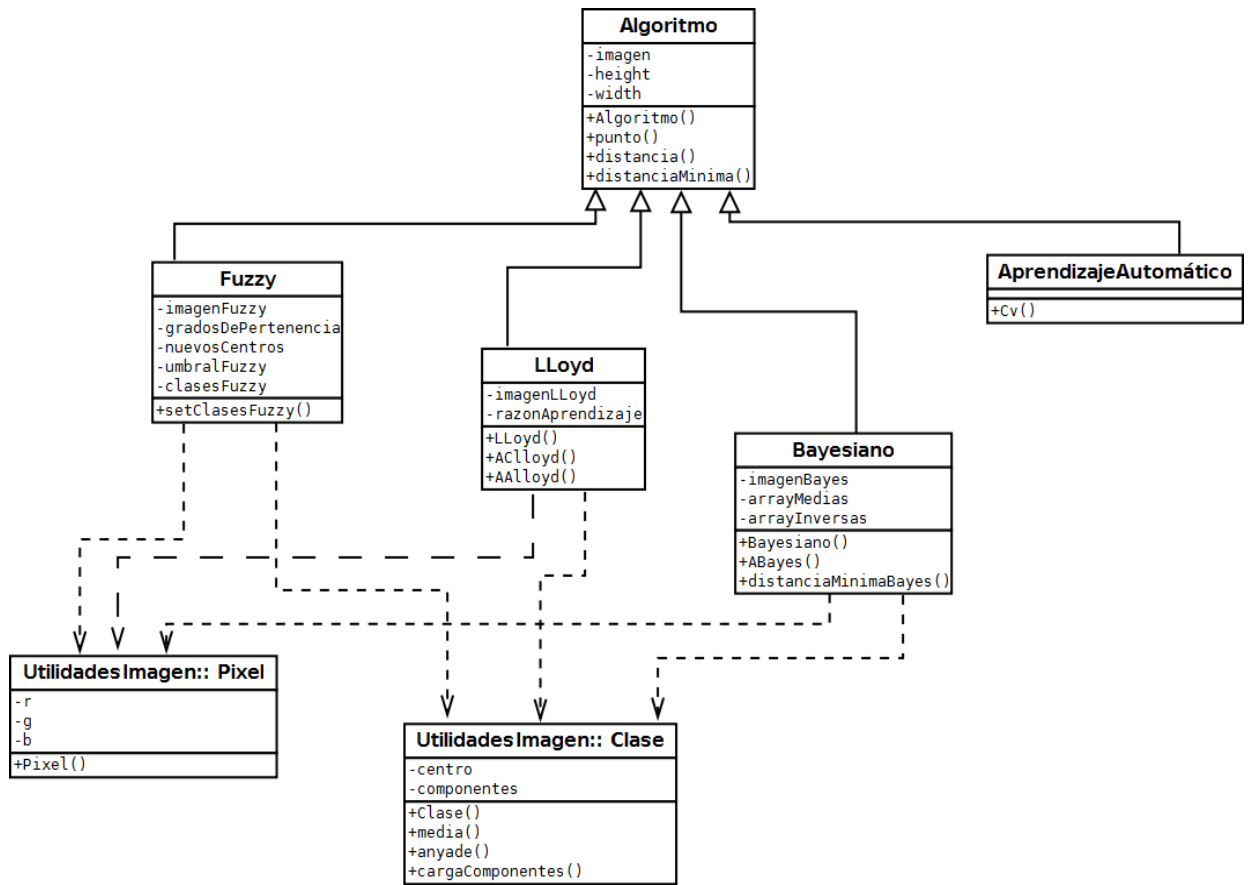


Figura 4. Diagrama de clases del paquete Algoritmo

### 5.3 DIAGRAMAS DE SECUENCIA.

A continuación se detalla el funcionamiento de los dos casos de uso fundamentales en nuestra aplicación como son entrenamiento y clasificación. Para el caso de uso de clasificación se utiliza como ejemplo el algoritmo de Agrupamiento Borroso (Fuzzy), con los algoritmos Lloyd y Bayesiano el funcionamiento es análogo.

#### Caso de uso 1: Entrenamiento.

En la figura 5 se detalla la interacción de objetos para el caso de uso “Entrenamiento”.

En la primera ejecución de la aplicación o cuando el usuario lo desee, selecciona una imagen, pulsa en el botón “entrenamiento” e introduce el umbral que le solicita la aplicación. A través de la clase TratamientoImagen se crea la matriz de píxeles que representa la imagen, se devuelve esta imagen al interfaz, para poder pasarla como argumento en la llamada a la clase AprendizajeAutomático, que es la que implementa la funcionalidad del algoritmo cuantización vectorial, algoritmo que nos devuelve la clasificación de clases (indicando centros de clase y componentes de la misma). Esta clasificación es la que guardamos en un fichero identificándolo con su número de umbral.

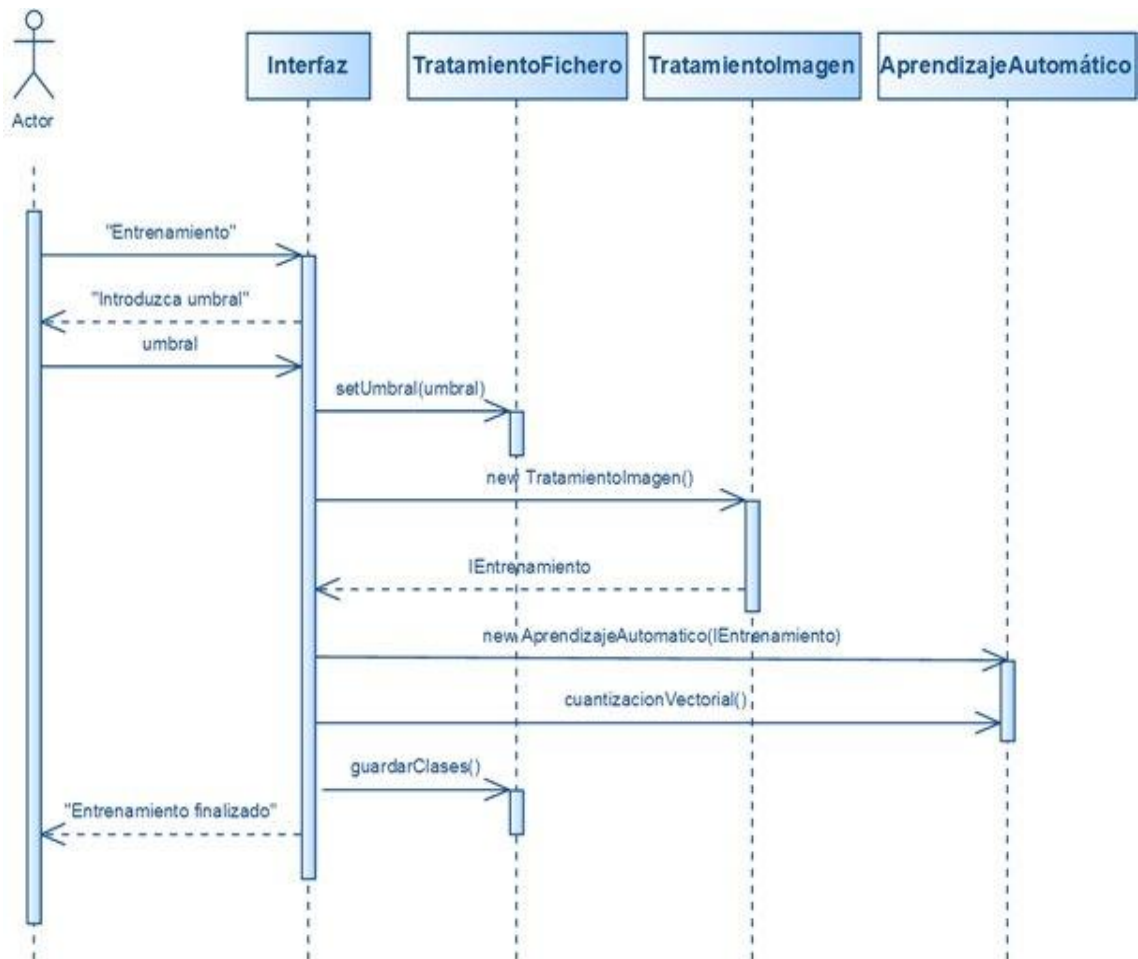


Figura 5. Diagrama de secuencia del caso de uso Entrenamiento.

### Caso de uso 2: Clasificación.

En la figura x se detalla la interacción de objetos para el caso de uso “Clasificación” para el caso de algoritmo de Agrupamiento Borroso.

El usuario carga una imagen y selecciona “clasificación Fuzzy”. La aplicación le solicita un umbral, y a través de la clase TratamientoImagen se genera la matriz de píxeles que representa la imagen de la misma forma que en el caso de uso anterior.

Esa matriz de píxeles es la que utiliza la clase Fuzzy para generar una clasificación, dicha clasificación se devuelve al interfaz en forma de matriz de píxeles, en el interfaz se realiza la asociación de cada clase con su etiqueta de color y la presentación de la imagen resultado.

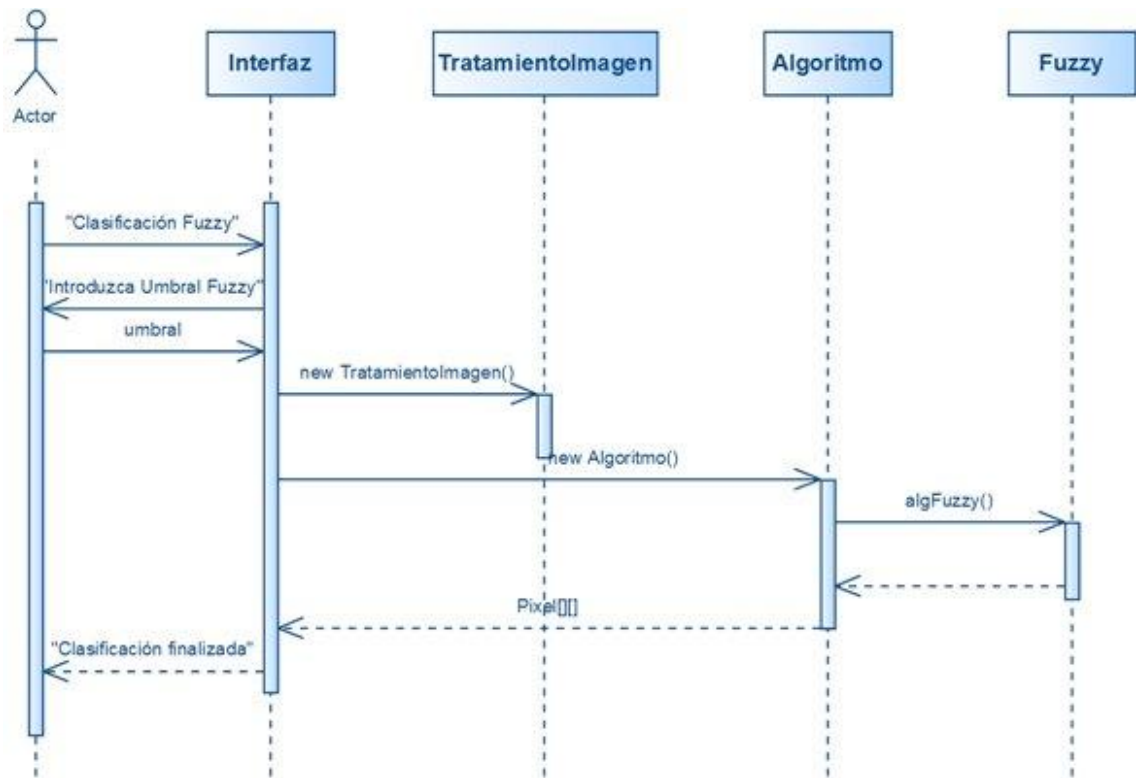


Figura 6. Diagrama de secuencia del caso de uso Clasificación.

## 6 RESULTADOS DEL PROYECTO

A continuación, se muestran resultados de dos ejecuciones (con distinto umbral) tanto para la fase de entrenamiento como de clasificación para una imagen, que actúa de testigo.

### 6.1 EJECUCIÓN 1

#### **Entrenamiento:**

Entrenaremos el sistema con la siguiente imagen en formato “bmp”. Como puede observarse, en ella se observan diferentes texturas correspondientes tanto a campos de cultivo como zonas boscosas o edificaciones y caminos rurales.

Para el entrenamiento mediante el algoritmo cuantizador vectorial elegimos un umbral de 125 para la determinación de las clases y sus centros. Dicho umbral es el definido en la sección 3.3.1.



Figura 7. Imagen para entrenamiento

Los resultados obtenidos durante el entrenamiento son los centros de las clases, que se obtienen mediante la aplicación del cuantizador vectorial. Estos centros son los que se muestran a continuación

$v_1$ : [106, 113, 75];  $v_2$ : [41, 37, 23];  $v_3$ : [188, 160, 130];  $v_4$ : [237, 202, 193]

### **Clasificación:**

Clasificaremos la siguiente imagen (bmp) con cada uno de los tres algoritmos explicados anteriormente:



Figura 8. Imagen original

### ***Algoritmo de Bayes:***

La Figura muestra el resultado de clasificar la imagen de la Figura 8 con el algoritmo de Bayes. En ella se distinguen claramente cuatro colores o etiquetas correspondiendo cada uno de ellos a una clase.

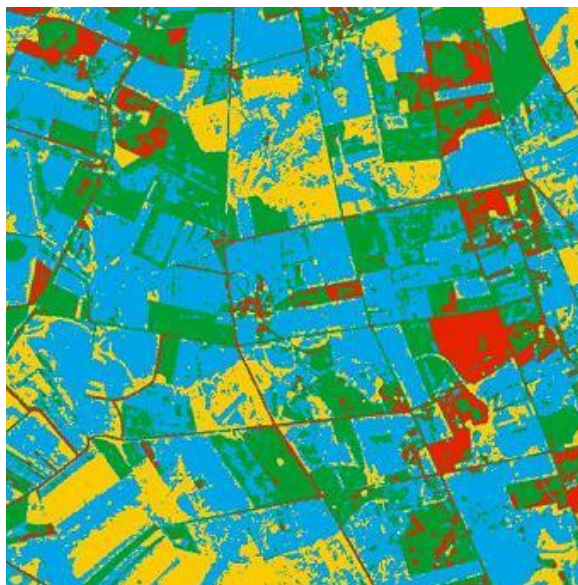


Figura 9. Resultados de la clasificación con Bayes

Los centros obtenidos son:

$\nu_1$ : [106, 113, 75];  $\nu_2$ : [41, 37, 23];  $\nu_3$ : [188, 160, 130];  $\nu_4$ : [237, 202, 193]

***Algoritmo agrupamiento borroso:***

El umbral utilizado es de 0.5. De esta forma, la siguiente imagen muestra el resultado de la clasificación. Como en el caso anterior, se diferencian claramente cuatro clases, una por cada color.



Figura 10. Resultados de la clasificación con Agrupamiento Borroso

Los centros obtenidos de la clasificación con agrupamiento borroso son:

$\nu_1$ : [104, 110, 73];  $\nu_2$ : [44, 38, 23];  $\nu_3$ : [181, 155, 124];  $\nu_4$ : [237, 202, 193]

***Algoritmo de Lloyd:***

Para la ejecución de este algoritmo se ha utilizado una razón de aprendizaje de 0.04. La razón de aprendizaje es la explicada en la sección 3.1.2. El resultado de la clasificación con este algoritmo y esta razón de aprendizaje son los reflejados en la Figura 11. Se diferencian cuatro clases.



Figura 11. Resultados de la clasificación con el algoritmo de Lloyd

Los centros obtenidos son:

$v_1$ : [80, 89, 50];  $v_2$ : [27, 24, 14];  $v_3$ : [128, 130, 92];  $v_4$ : [202, 174, 146]

## 6.2 EJECUCIÓN 2

### **Entrenamiento:**

Aprovecharemos las clases y los centros obtenidos durante el proceso del punto 6.1 para llevar a cabo la siguiente clasificación:

### **Clasificación:**

Como en el caso anterior, clasificaremos la siguiente imagen (bmp) con cada uno de los tres algoritmos explicados anteriormente:



Figura 12. Imagen original

***Algoritmo de Bayes:***

El resultado de la clasificación con el algoritmo bayesiano es la mostrada en la Figura 13. De nuevo se observan cuatro clases.

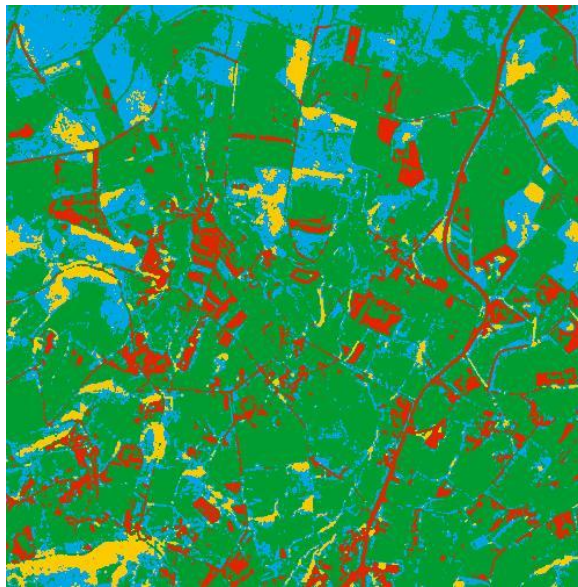


Figura 13. Resultados de la clasificación con el algoritmo de Bayes.

Los centros obtenidos son:

$\nu_1$ : [80, 89, 50];  $\nu_2$ : [27, 24, 14];  $\nu_3$ : [128, 130, 92];  $\nu_4$ : [202, 174, 146]

***Algoritmo de Agrupamiento Borroso:***

El umbral utilizado es de 0.5. De esta forma, vemos el resultado devuelto el sistema en la Figura 14:

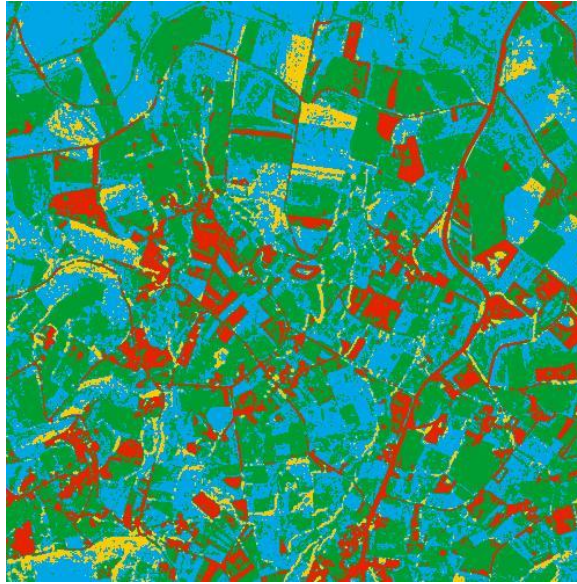


Figura 14. Resultados de la clasificación con Agrupamiento Borroso

El sistema devuelve los siguientes valores de los centros:

$\nu_1$ : [80, 95, 55];  $\nu_2$ : [27, 33, 11];  $\nu_3$ : [122, 133, 89];  $\nu_4$ : [202, 183, 155]

***Algoritmo de Lloyd:***

Para la ejecución de este algoritmo se ha utilizado una razón de aprendizaje de 0.04. Dicha razón queda definida en la sección 3.1.2.

A continuación, se muestra el resultado en la Figura 14. Quedan bien diferenciadas cuatro clases, cada una representada por un color:

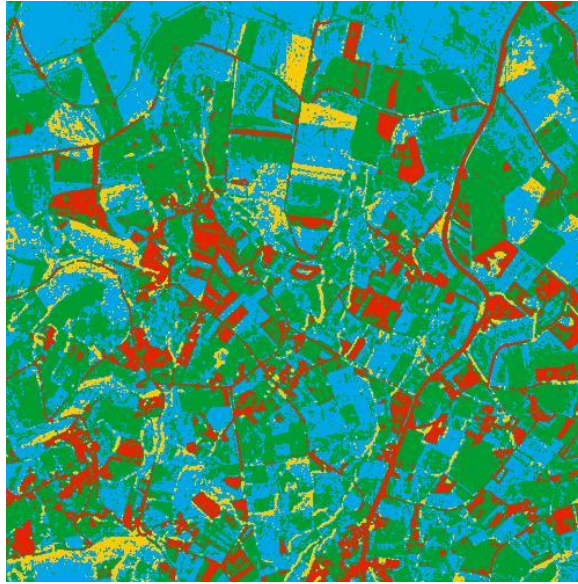


Figura 15. Resultados de la clasificación con el algoritmo de Lloyd

Los centros obtenidos al clasificar la imagen de la Figura 15 con el algoritmo de Lloyd y una razón de aprendizaje de 0.04 son:

$$\mathbf{v}_1: [97, 119, 75]; \mathbf{v}_2: [26, 42, 13]; \mathbf{v}_3: [149, 152, 110]; \mathbf{v}_4: [243, 237, 227]$$

## 7. BIBLIOGRAFÍA.

- 1) G. Pajares y J.M. de la Cruz, *Visión por Computador: Imágenes digitales y Aplicaciones*, RA-MA, 2001.
- 2) Zimmerman H.J 1991. *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, Norwell.

# ANEXO 1

## MANUAL DE USUARIO:

### ***Índice:***

1- Cargar una imagen.....	50
2- Entrenar el sistema.....	53
2.1- Entrenar el sistema desde cero.....	53
2.2- Utilizar un entrenamiento anterior.....	56
3- Guardar el entrenamiento .....	58
4- Clasificación de una imagen .....	59
5- Guardar una imagen .....	61

Mediante este anexo se trata de ilustrar al usuario sobre los pasos a seguir para conseguir una correcta utilización de la aplicación.

## 1. Cargar una imagen

Antes de realizar cualquier imagen en el sistema es necesario cargar una imagen con la que trabajar. Para ello, mostramos los pasos a seguir:

*“Archivo” -> “Cargar”*

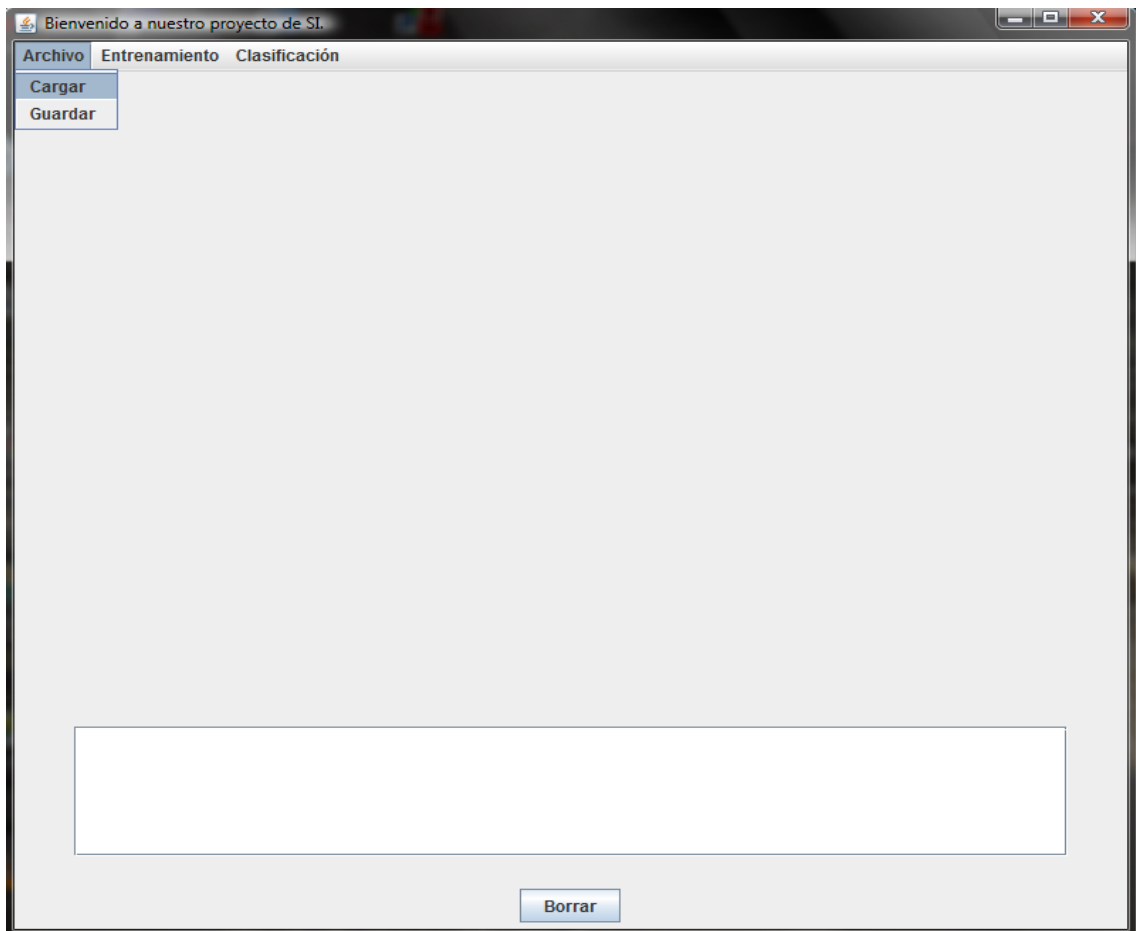


Figura A1. Pantalla de inicio de la aplicación

A continuación se selecciona la ruta de la imagen, tal y como se muestra en la imagen de la figura A2. En ella se puede observar cómo aparece una ventana para seleccionar la ruta de acceso a la imagen que se desea cargar para su procesamiento:

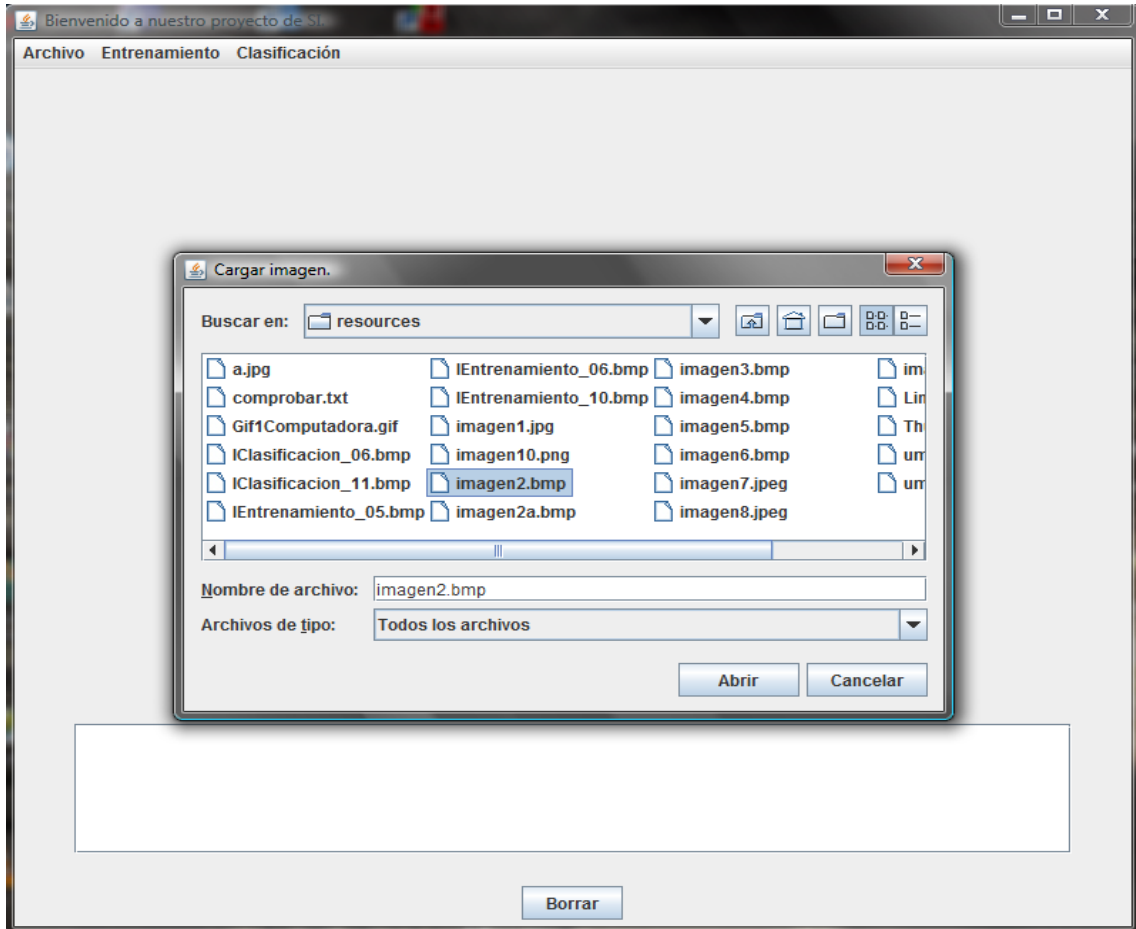


Figura A2. Selección de imagen

Una vez seleccionada la imagen que deseamos, pulsaremos el botón ***“Abrir”***.

El resultado es el mostrado en la Figura A3. En este momento se tiene la imagen cargada en el interfaz y le aplicación queda preparada para ser entrenada con esta imagen o para clasificar dicha imagen.

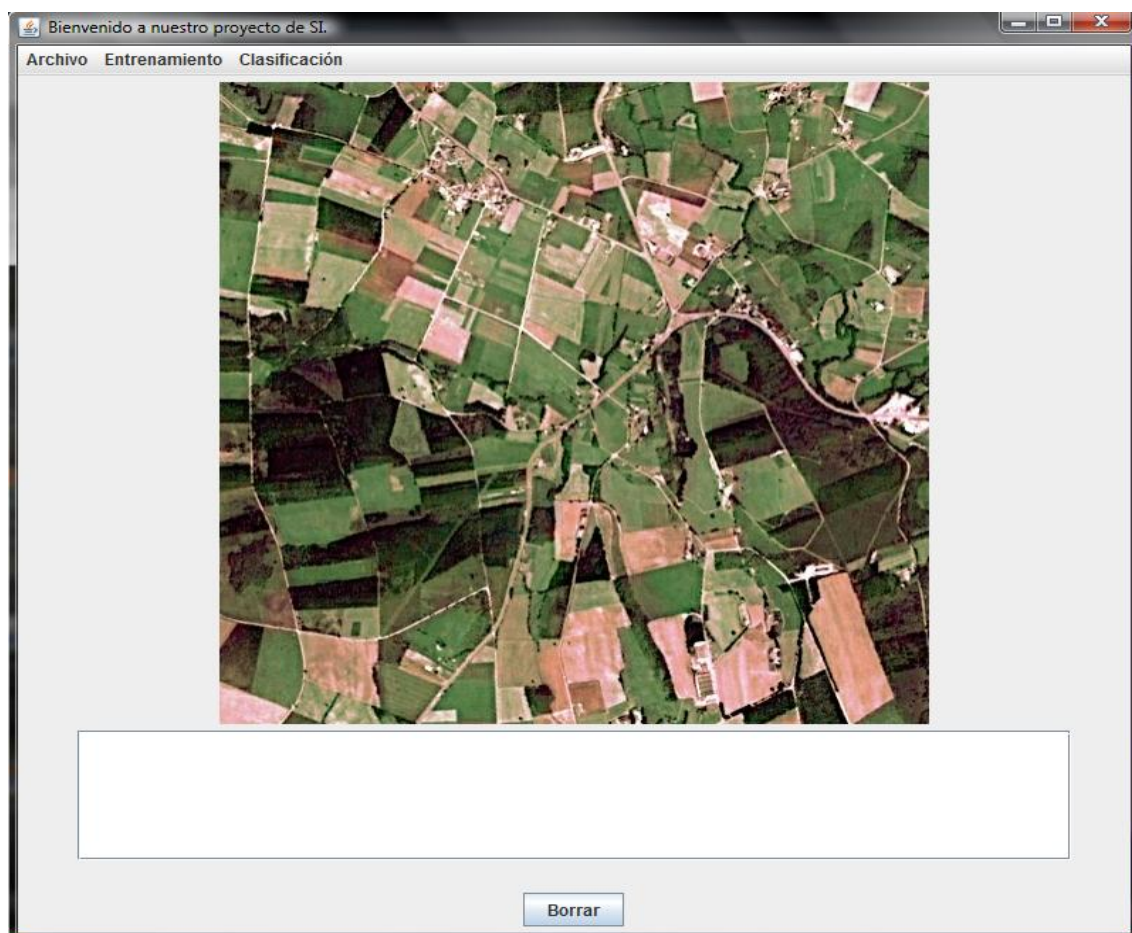


Figura A3. Imagen cargada en el interfaz

## 2. Entrenar el sistema

### 2.1 Entrenar el sistema desde cero

Para usar el sistema sin utilizar ningún entrenamiento previo es necesario cargar una imagen siguiendo los pasos explicados en 1 y a partir de entonces:

***“Entrenamiento”->“Entrenar la imagen”***

La Figura A4 muestra el menú de entrenamiento desplegado con la opción de entrenar la imagen seleccionada.

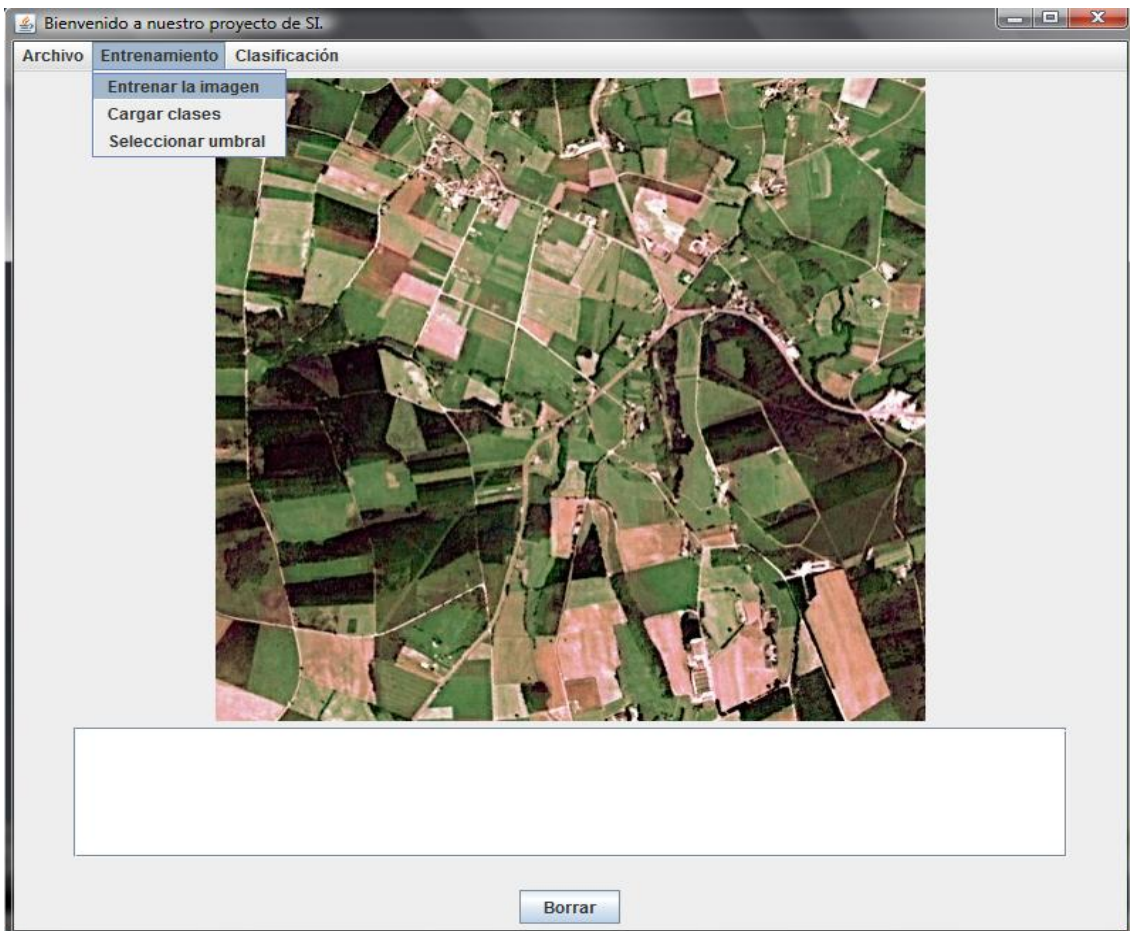


Figura A4. Menú de entrenamiento

A continuación, introduciremos el umbral con el que deseamos realizar el entrenamiento en el cuadro de texto que se muestra en la Figura A5.

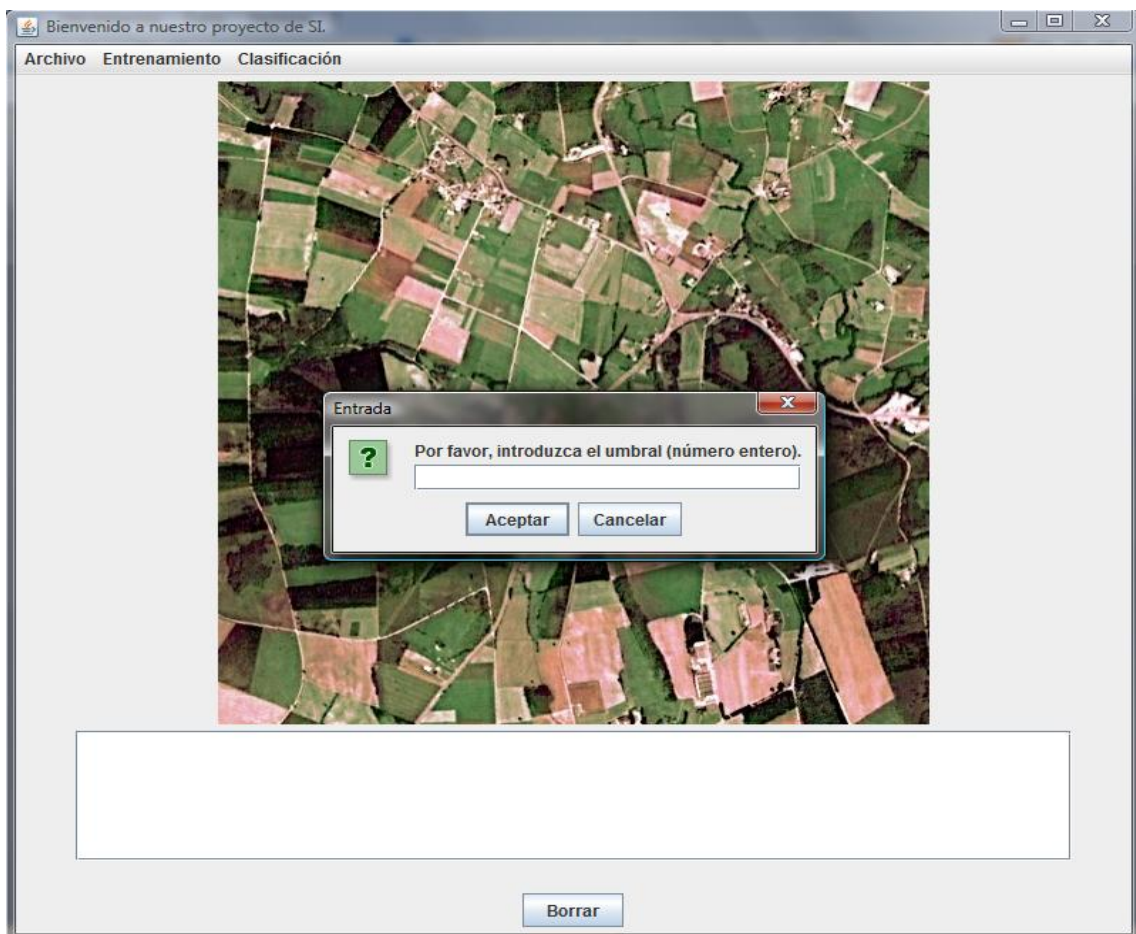


Figura A5. Pantalla de introducción del umbral de entrenamiento

Una vez acabado el entrenamiento, el resultado es el mostrado en la Figura A6. En ella se puede observar el valor de los centros de las clases.

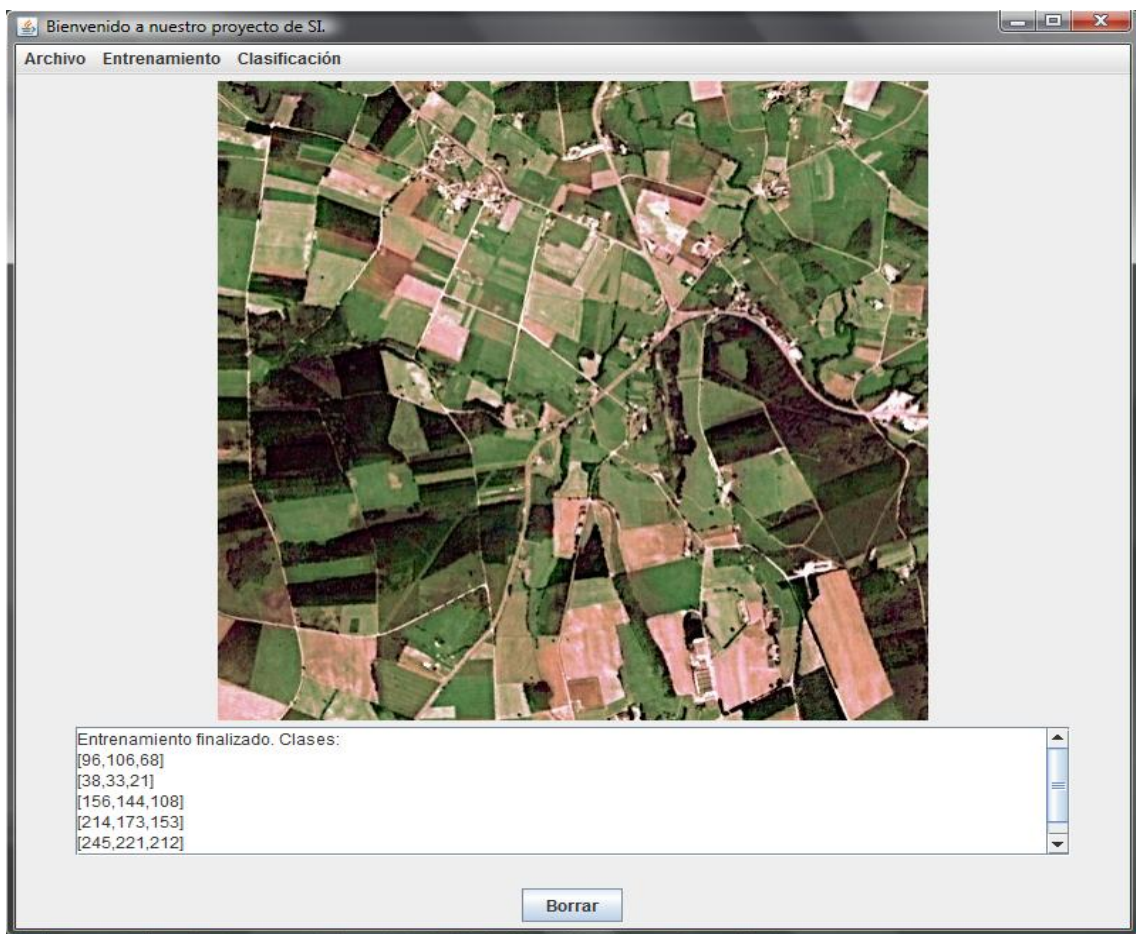


Figura A6. Pantalla que muestra el resultado del entrenamiento

## 2.2 Utilizar un entrenamiento anterior

En este caso, se puede cargar una imagen para continuar con el entrenamiento o para clasificar directamente con las clases que cargamos.

La imagen se carga como en el punto uno de este apéndice, con la particularidad de que no es necesario que esté cargada (aunque puede estarlo) en el momento de cargar las clases.

### *“Entrenamiento”->“Cargar clases”*

La Figura A7 muestra el menú Entrenamiento desplegado con la opción de cargar clases seleccionada.

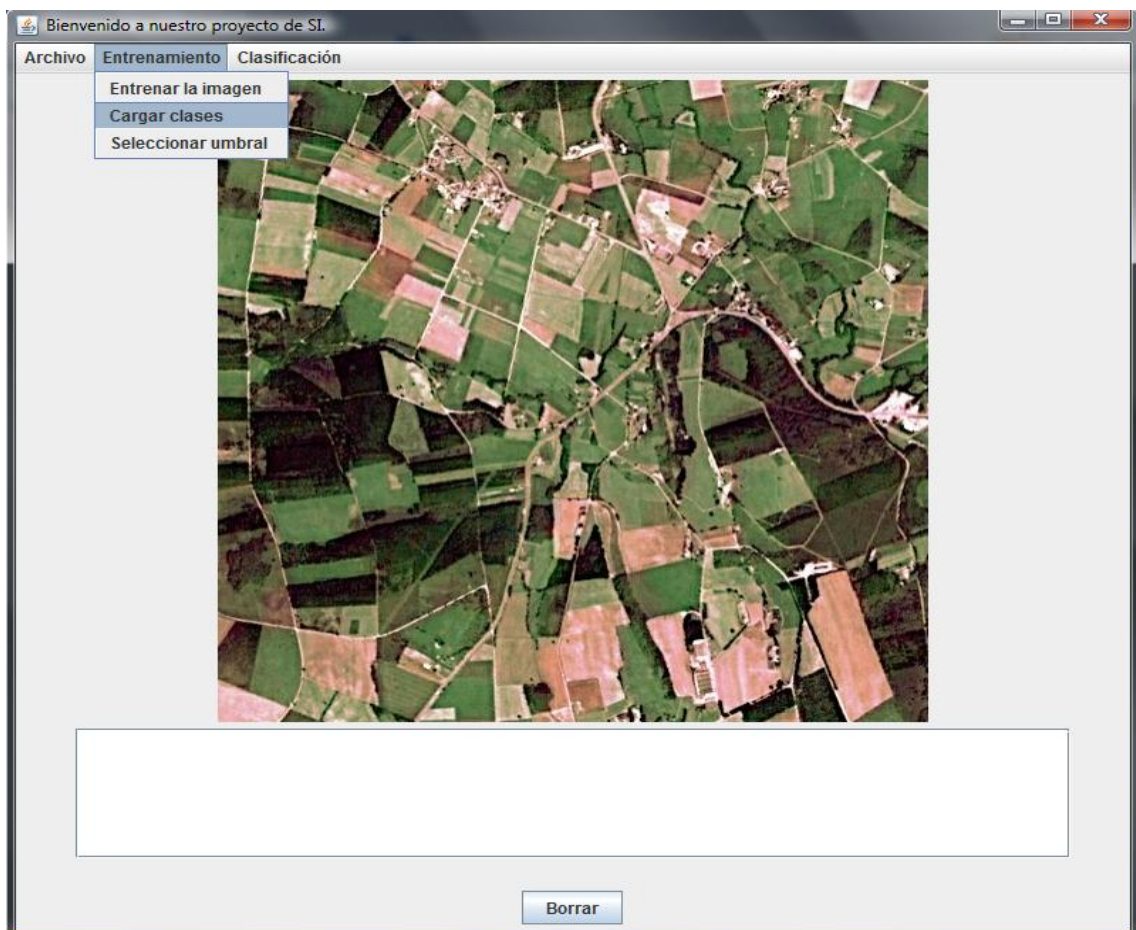


Figura A7. Menú de entrenamiento

Una vez cargadas las clases el programa nos informa, como se muestra en la Figura A8.

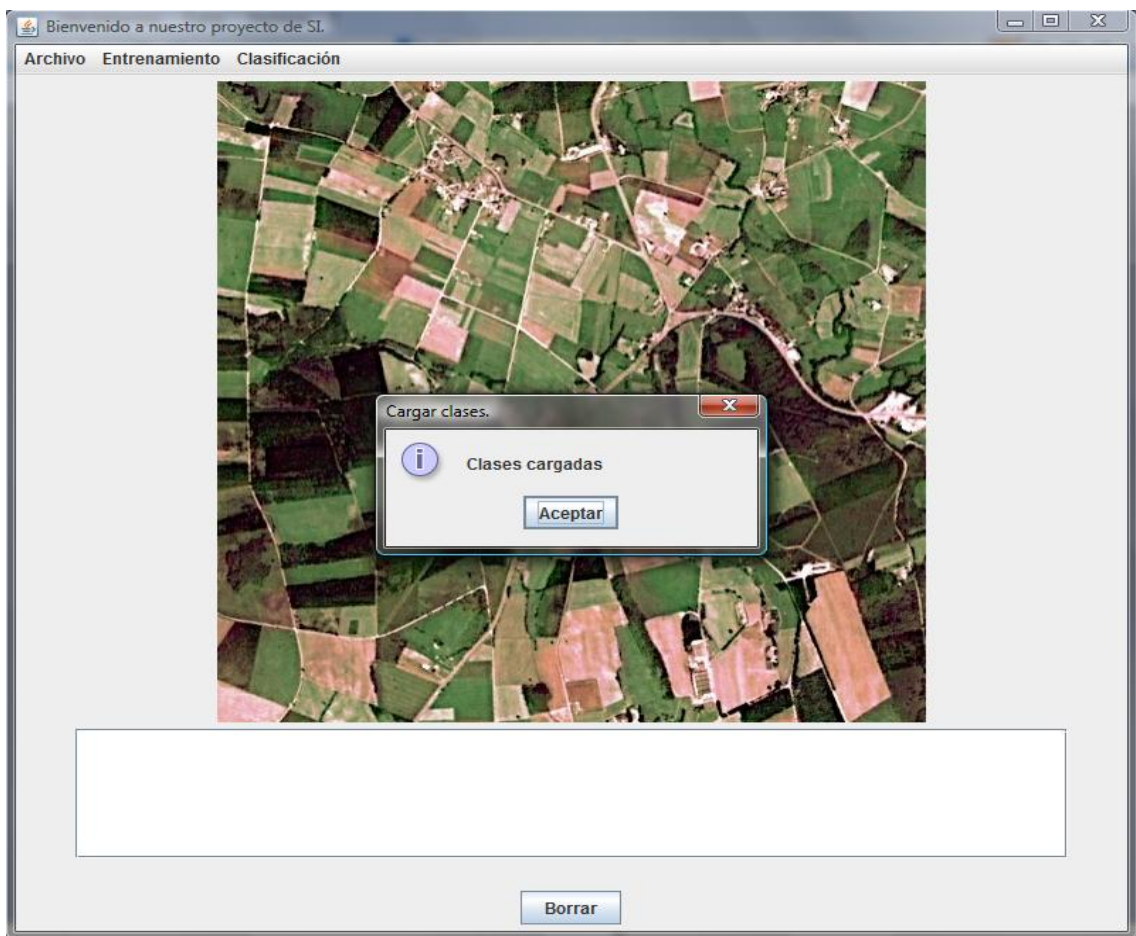


Figura A8. Información de clases cargadas.

### 3. Guardar el entrenamiento

Los resultados obtenidos en cada entrenamiento se guardan automáticamente al salir de la aplicación y al cambiar de umbral para realizar un nuevo entrenamiento.

El archivo en el que se salvan los resultados tiene el nombre de :

umbral +(número del umbral) + .bin

Por ejemplo, si hemos entrenado el sistema con un umbral de 150, los resultados se guardarán en el archivo ***“umbral100.bin”***.

Dicho fichero está alojado en el directorio del programa en la carpeta ***“resources”***.

## 4. Clasificación de una imagen

Para clasificar las texturas de una imagen necesitamos tener la imagen cargada previamente (punto 1) y haber entrenado el sistema o tener cargadas las clases de un entrenamiento anterior (punto 2).

En este manual realizaremos la clasificación de una imagen con el algoritmo de Bayes. Para proceder con el algoritmo de Agrupamiento Borroso o de Lloyd, el proceso es análogo.

### *“Clasificación”->“Algoritmo de Bayes”*

La Figura A9 muestra el menú Clasificación desplegado con la opción Algoritmo de Bayes seleccionada.

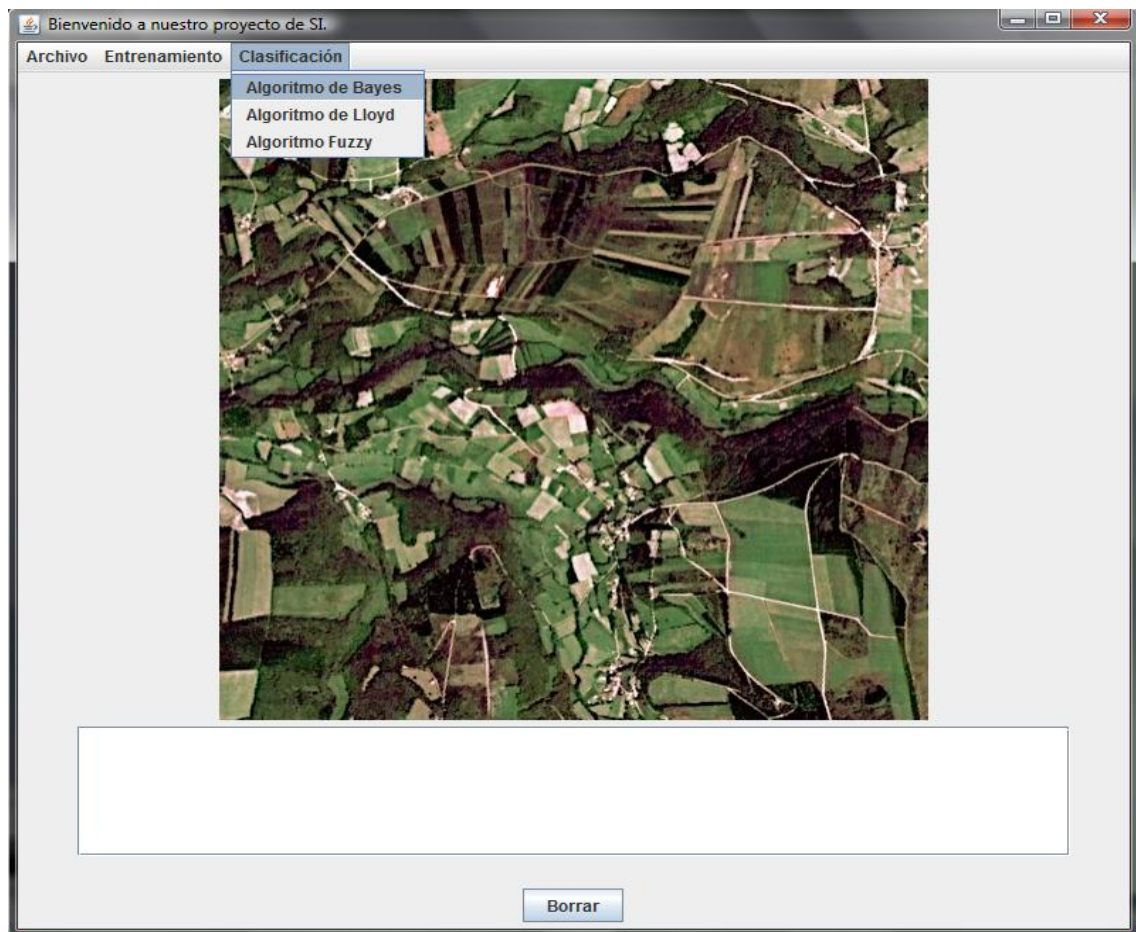


Figura A9. Menú de clasificación

Una vez terminada la clasificación, tanto la imagen resultado como el número de las clases y sus centros se muestran en la Figura A10.

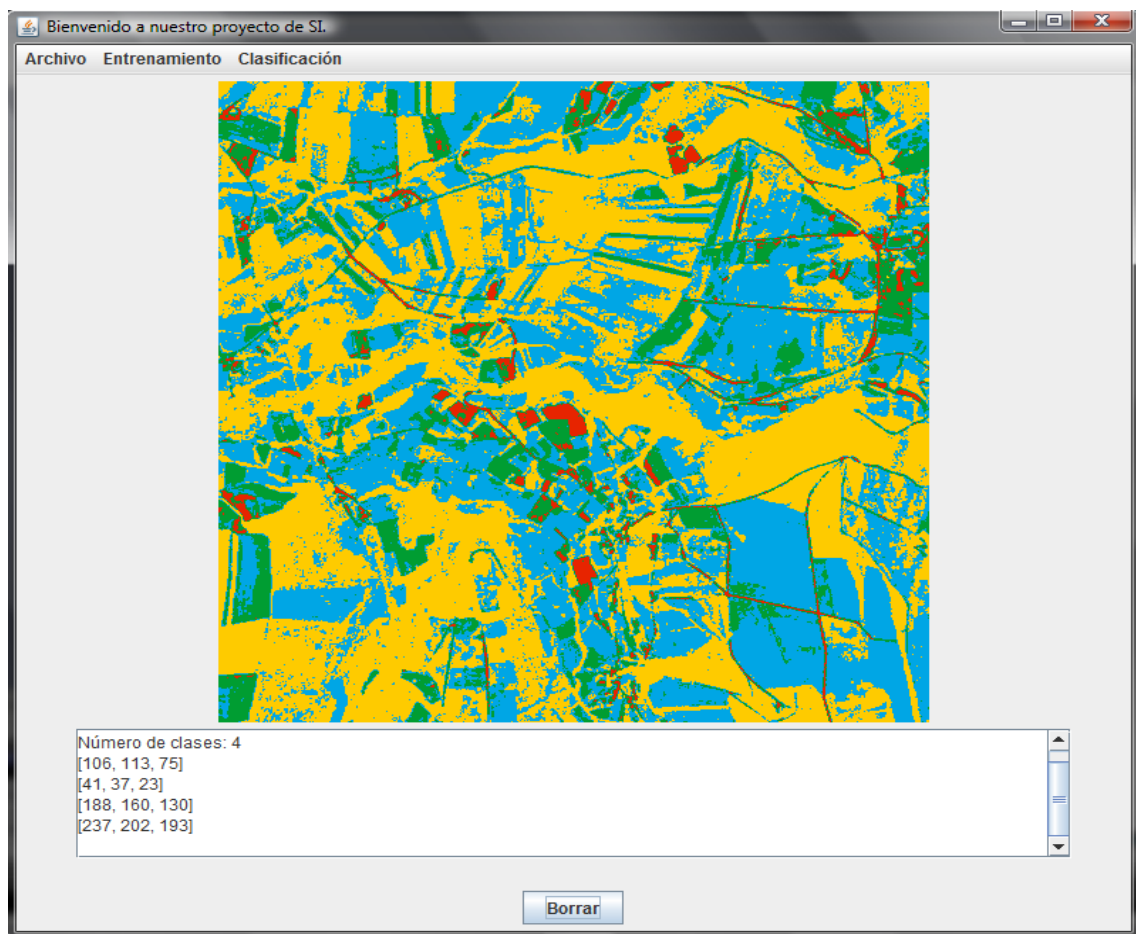


Figura A10. Resultado de la clasificación

## 5. Guardar una imagen

Una vez realizada la clasificación, el sistema nos ofrece la posibilidad de guardar la imagen generada, para ello es necesario seguir los siguientes pasos:

***“Archivo”-> “Guardar”***

La Figura A11 muestra el menú archivo desplegado con la opción Guardar seleccionada.

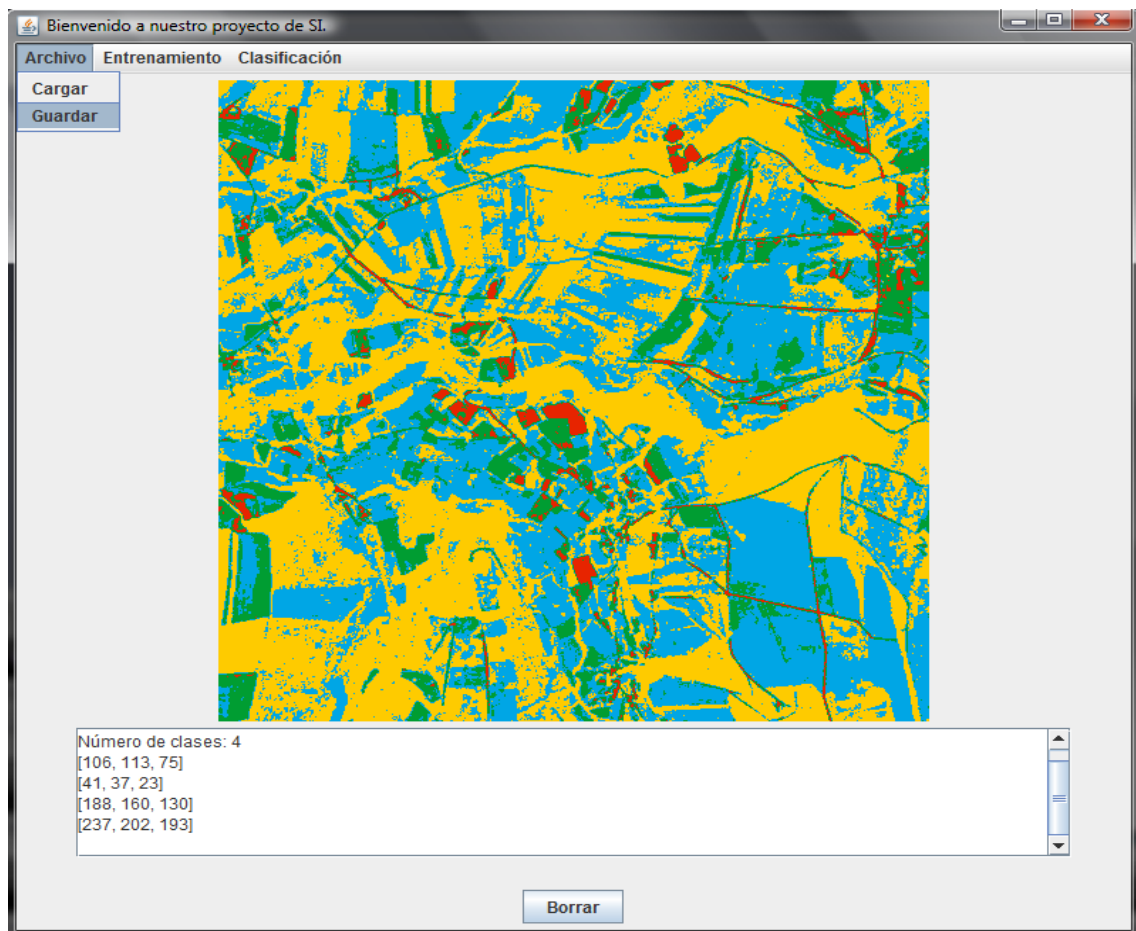


Figura A11. Menú Archivo desplegado

A continuación seleccionamos la ruta en la que queremos guardar la imagen y le damos un nombre (con extensión):

En la Figura A12 se muestra una pantalla en la que se puede seleccionar la ruta para guardar la imagen clasificada.

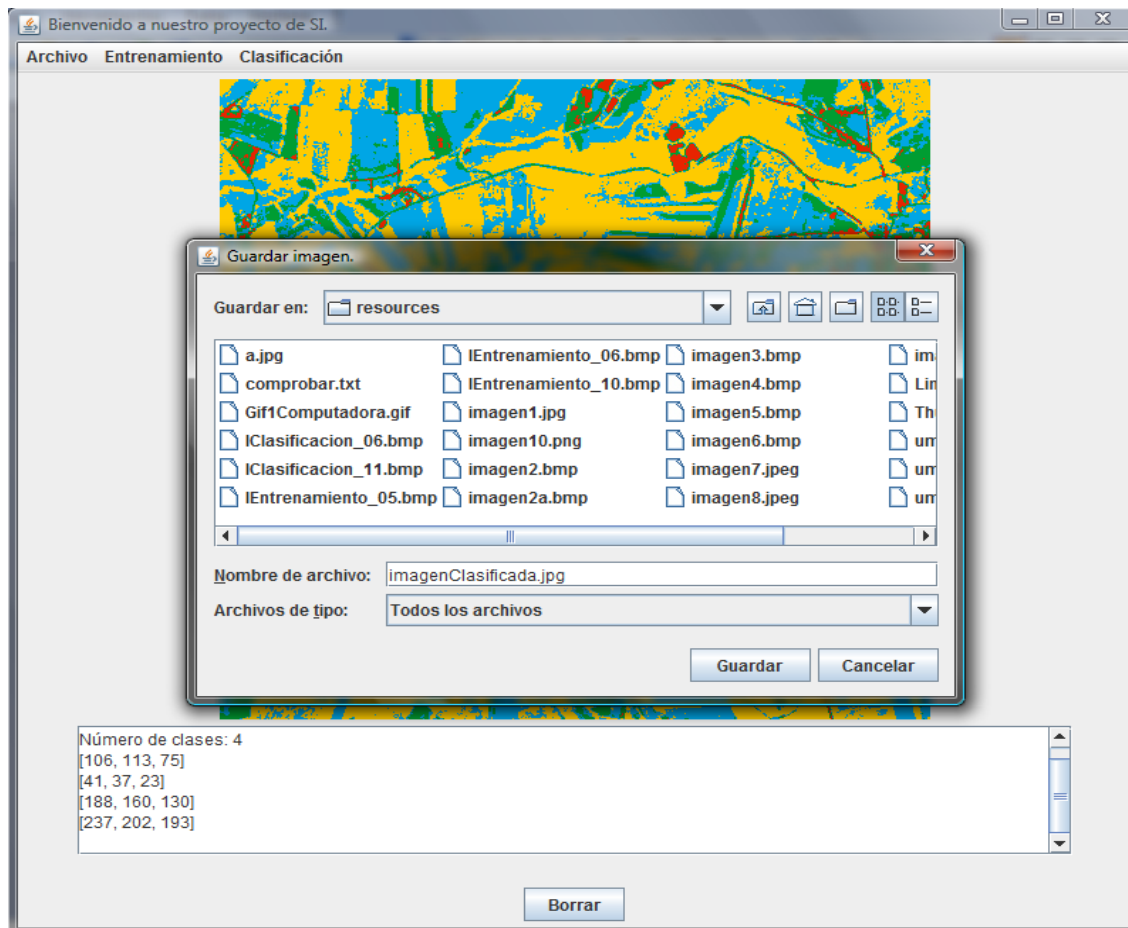


Figura A12. Selección de ruta para guardar la imagen

Para finalizar, pulsamos el botón **“Guardar”**.