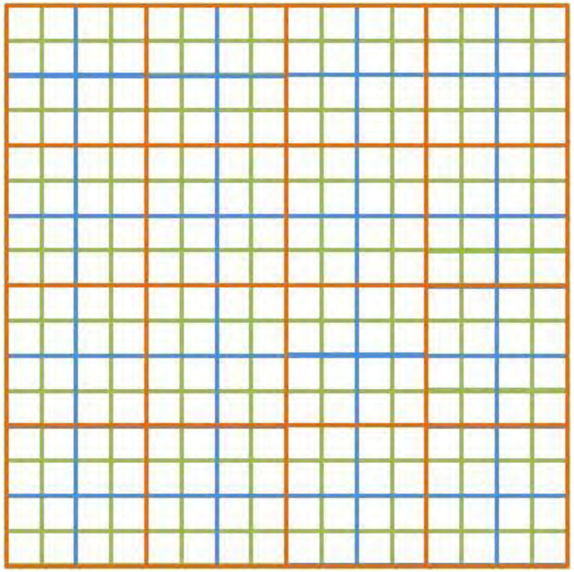
An integer nested grid for INSPIRE Orthoimagery and Elevation themes



Guillermo Villa, Jose Antonio Tejeiro, Juan Carlos Ojeda, Borja Rodríguez, (Instituto Geográfico Nacional. Spain),
David Hernández (Universidad de Castilla la Mancha. Spain)
Xalo Fernández (Ministerio de Agricultura, Alimentación y Medio Ambiente. Spain)

Inspire Conference 2016. Barcelona. Spain

1. Motivation

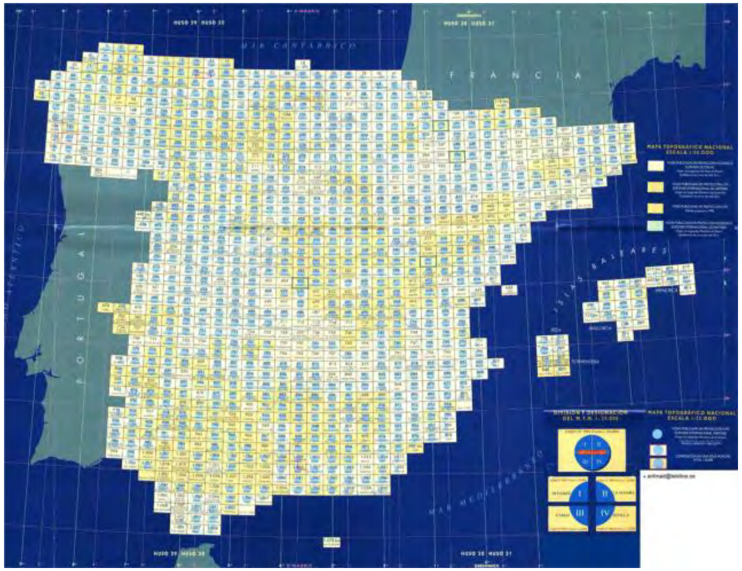
- INSPIRE Orthoimagery theme includes two different types of data with slightly different needs:
  - Aerial orthophotos
  - Satellite orthoimages

- Usual workflows for production, archiving, dissemination and use of both kind of orthoimages are very inefficient

- In this presentation we propose the use of a unique "nested grid" for the production, storage, processing and dissemination of orthoimages and also for other raster data, such as biophysical parameters, Digital Elevation Models, raster maps, etc.

2. Usual workflows in the production and disemination of Orthoimages

1) Produce uncompressed orthophotos, by orthorectifiying aerial images in "production units" normally called "sheets" for historical reasons. (e.g: a single GeoTIFF file)

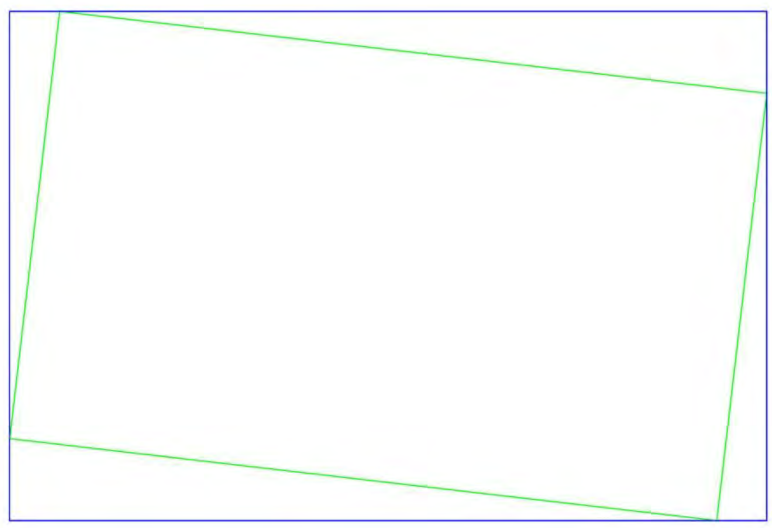


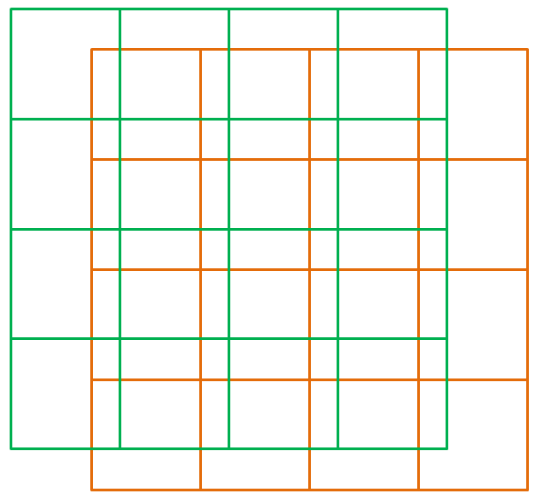
2) Mosaic several uncompressed orthophotos in one larger compressed image file (e.g: in JPEG2000 format)

3) Set up WMS and WCS services

4) Produce JPEG tiles for "cached" WMTS services

Problem 1: Non-alignment of pixels



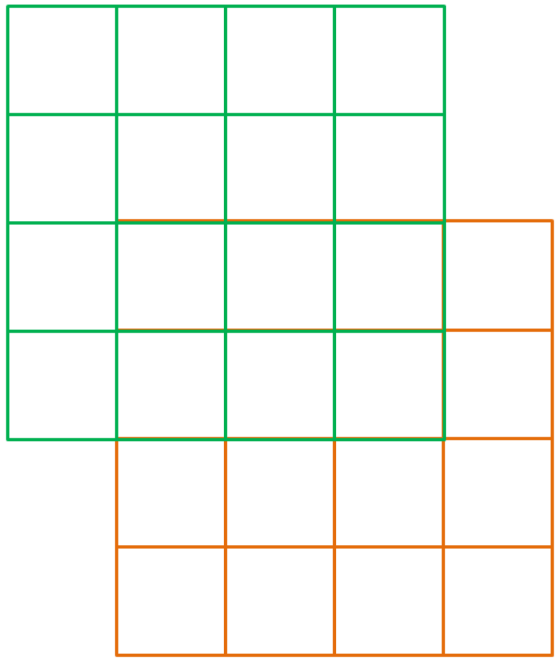


- If we take the strict bounding rectangles as limits for the orthoimages, adjacent sheets will have “nonaligned” pixels because the upper left corners of these orthoimages will not be multiplse of pixel size

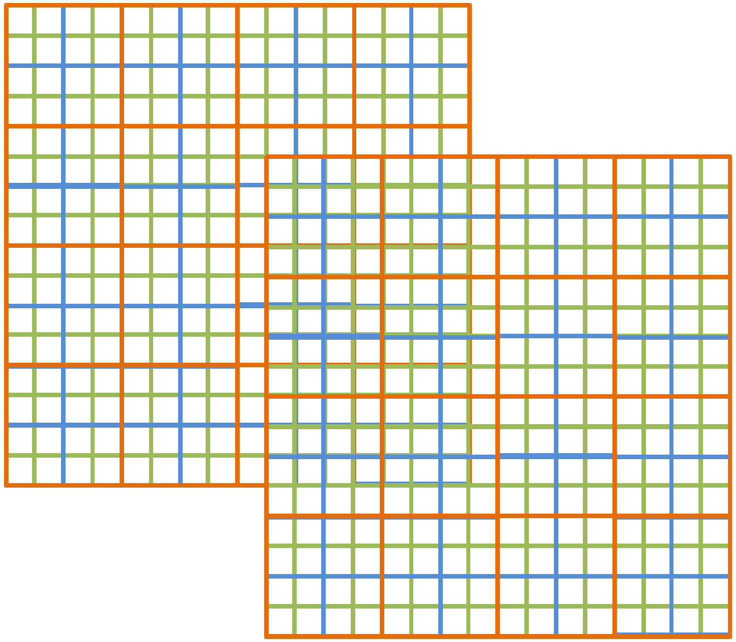
- This makes it impossible to mosaic multiple orthophotos or even overlay them in a viewer unless we resample them

- But resampling is computing demanding and causes image degradation....

Forcing the alignment of pixels

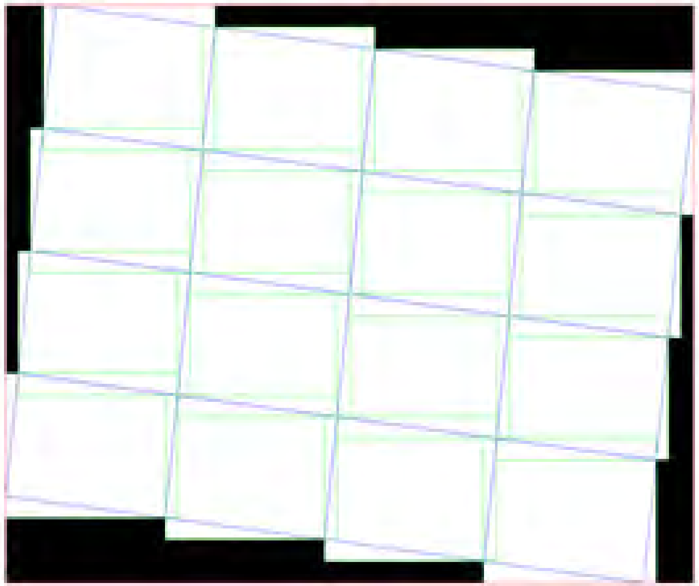


… so we should “force” the alignment of the pixels by making (X, Y) coordinates of the “upper left corner” exact multiples of pixel size



- But even if we “force” the alignment of the pixels at the original GSD (green pixels of LOD=n are aligned) , it does not ensure pixel alignment in the next levels of the pyramid
- In these level and the next ones it is impossible to mosaic multiple orthos, or display the “virtual mosaic” without resampling them
- This hampers greatly multirresolution and multi-temporal processing

Problem 2: Empty wedges



- When we mosaic several uncompressed orthophotos in one larger compressed image file (in order to facilitate management and dissemination, by reducing the number of files) empty wedges appear
- These "null" pixels cause a lot of problems afterwards

Problem 3: Fuzzy borders

When we reproject an orthoimage to a different UTM zone, the borders of the resulting borders have "intermediate" values due to resampling, so they cannot be easily eliminated

Problems 4, 5,….

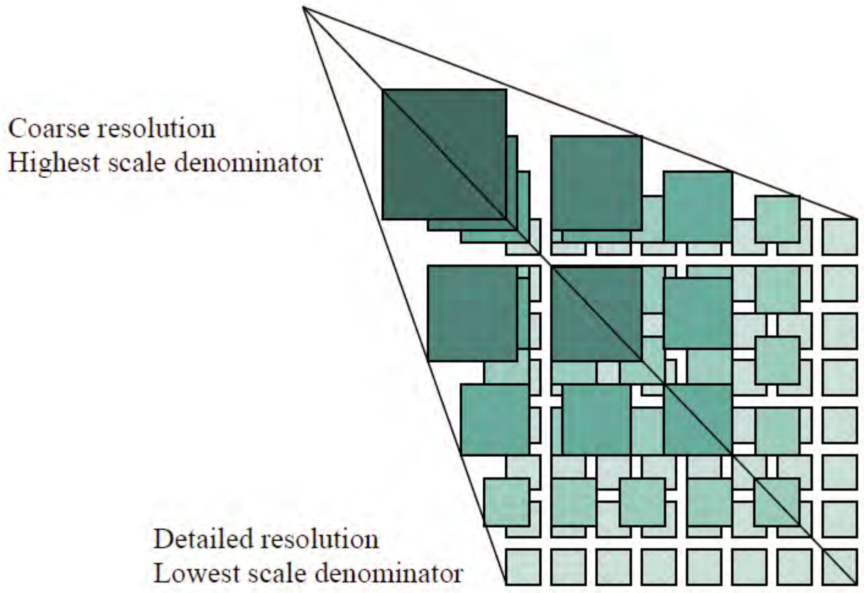
- Multiple compressions and decompressions (e.g: JPEG2000 + JPG Tiles) -> image degradation
- Multiple versions stored
- etc, etc…

3. Efficient visualization needs "cached" tiled services

- The most efficient web visualization of raster data is through "cached" "tiled web services" (e.g. WMTS)

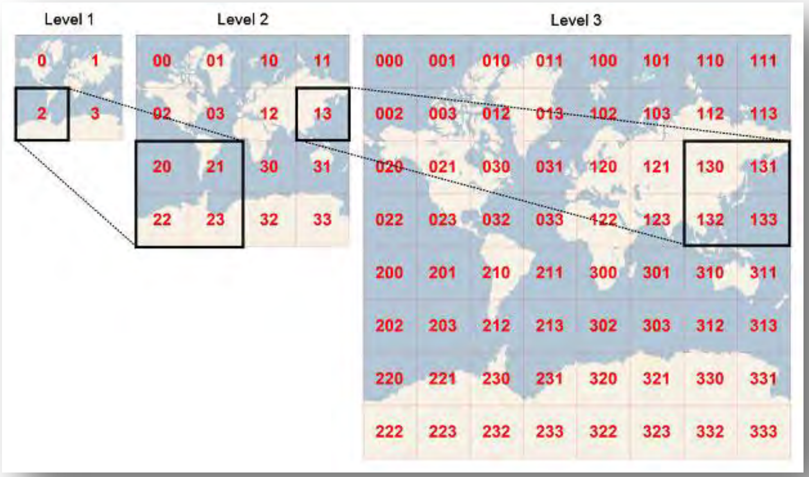
- The same is true for rasterized maps

- People is used to this performance and no longer accept the slowness of "non-tiled" services (e.g. WMS)



4. Interoperability of Map Projections

- Light web clients do not reproject nor resample on the fly, so in order to be able to overlap several web layers in one light web client, all data sources must be in the same projection and the have the same pixel sizes and positions



5. Efficient production

- It is much more efficient to produce raster datasets thinking on tiled services publication right from the beginning

- In this way, we avoid a lot of problems:
  - empty wedges
  - non aligned pixels
  - multiple reprojections
  - multiple resamplings
  - multiple compression and storage processes, etc.

6. Recommendations for an optimal workflow

The following recommendations appear for an optimal workflow:

1. Avoid the use of map projections with different zones (e.g. UTM)

2. Avoid repeated resampling: ideally only one resampling should be performed during the whole process

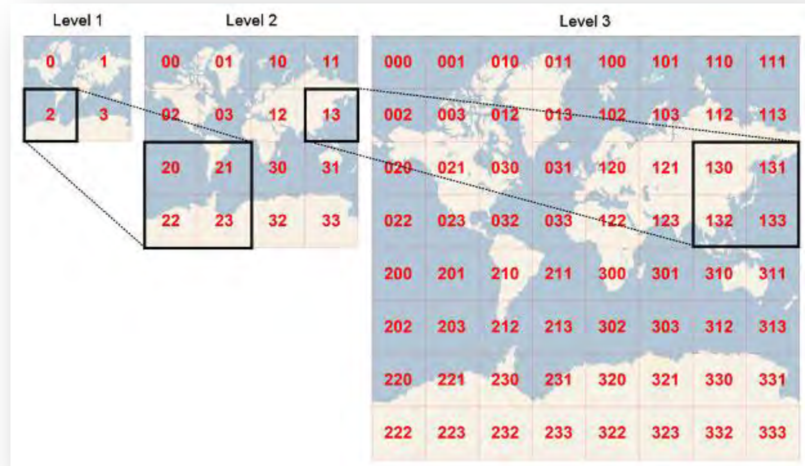
3. Pixel borders should be aligned at all levels of the pyramid

4. Avoid "empty wedges". Production "sheets" should be rectangles in the map projection and oriented to the North

5. Avoid repeated compressiosn and decompressions:
    - In the case of orthoimages only one compression and one decompression should be performed during the whole process
    - In remote sensing no compression should be applied

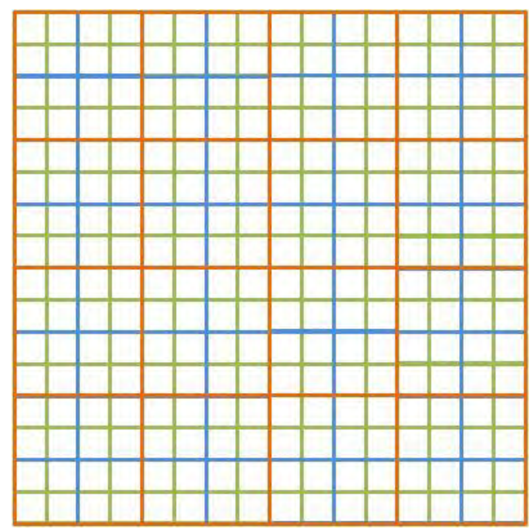
7. Tiling Schemas and nested grids

- A “tiling schema” is necessary to obtain a coherent multiresolution coverage of an area



- An optimal tiling schema should be a “nested grid”

- A “nested grid” is a space allocation schema that assures completely coherent and consistent multiresolution coverage of the whole working area by organizing image footprints, pixel sizes and pixel positions at all pyramid levels.

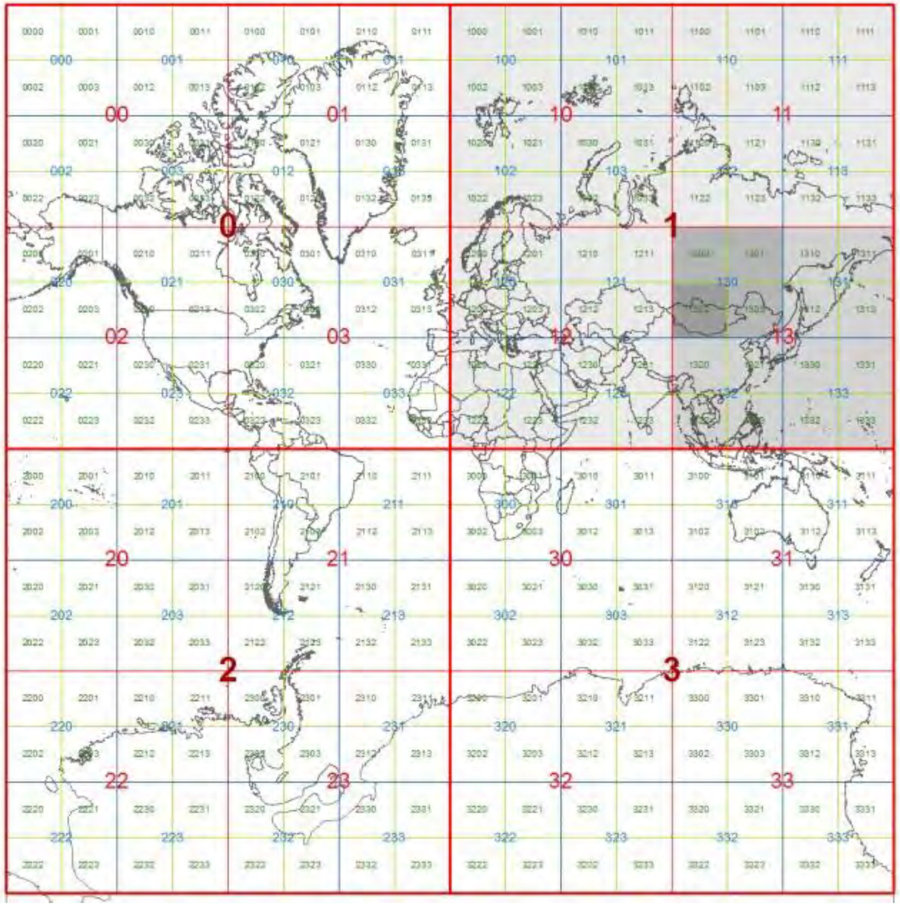


How to create a “nested grid”

- The only way to obtain a “nested grid” is to start with a single “tile” covering all the working area, and divide it iteratively in 2x2

- If we start with a tiles with a number of rowes and columns power of 2, the total numer of rows and columns is allways

  2^n= 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,…



- The map projections that allow to generate a nested grid are rectangular projections: all the working area (all the wold) in a single rectangle
- Frequently used map proyections (e.g: UTM, Lambert,…) are not rectangular so they should be discarded
- The two more frequently rectangular used map projections are:

1) Geographic *projection*. *It covers the whole Earth with one rectangle. Source: Wikipedia*

2) Mercator *projection. It covers the biggest part of the inhabited areas with one rectangle. Source: Wikipedia*

7. Web Mercator: “de facto” standard projection

- In the last times a “de facto” standard has emerged in web mapping: “Web Mercator” projection (EPSG:3857)

- It is used and supported by a great number of geospatial data and API providers (e.g. Google, Microsoft, ESRI,…) and very important Open Source and Open Data initiatives (Open Street Maps, Mapbox, etc.)

Web Mercator

The Standard for Sharing Data on the Web

Many organizations use geographic information system (GIS) data stored in a local projected coordinate system. When preparing to share maps or layers on the Web, it is recommended to reproject your source data to the Web Mercator coordinate system. Doing so will ensure that your map data is located correctly and aligns properly with other services such as popular content providers Microsoft® Bing™ Maps, Google Maps™, and ESRI® ArcGIS™ Online, which have standardized their services on the Web Mercator coordinate system.

Projecting Your Data to Web Mercator

In ArcGIS®, you have two options if you want to reproject your data to the Web Mercator coordinate system. The first, using the Project tool, creates a copy of your data that replaces the original coordinates in your source data with new coordinates from the Web Mercator coordinate system. The second, setting the ArcMap™ data frame coordinate system, instructs ArcGIS to calculate the new coordinates only when needed—leaving your source data unchanged. Since the first option requires a complete copy of your data to be made, you may want to use it for data that does not change very often—such as basemaps. You can then use the second option for data that changes often—when it would be too time-consuming to keep creating copies of your data. Both options are presented for your reference.

For ArcGIS Server Web mapping applications, developers can use the geometry service to reproject features before making area and length calculations. Reprojecting features is not a problem since positional accuracy is preserved when projecting between Web Mercator and other coordinate systems.

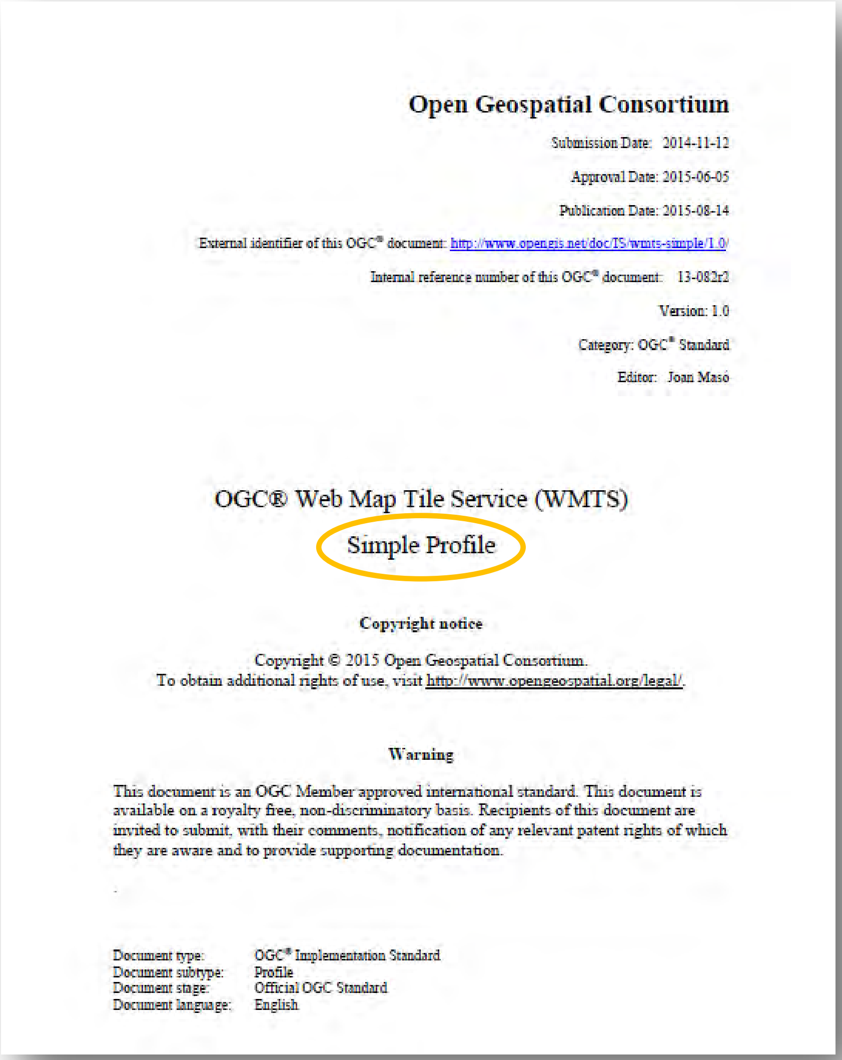
ESRI

The reasons for this massive adoption are multiple:

- Rectangular : allows the whole Earth (or the biggest part of inhabited areas) in one single square tile
- Almost conformal: locally maintains the shapes of objects that have a natural aspect at all latitudes (e.g: buildings, roundabouts)
- No different zones (one single projection)
- North is always straight up (Geographic North = Projection North)
- Very efficient computation (thanks to the auxiliary sphere: easier formulas)

8. OGC Standard: WMTS Simple Profile

- WMTS Simple Profile has recently been approved as official OGC standard.
- The objective is to solve the frequent incompatibilities between different implementations of WMTS standard.
- This tiling schema is a nested grid.


Open Geospatial Consortium

Submission Date: 2014-11-12

Approval Date: 2015-06-05

Publication Date: 2015-08-14

External identifier of this OGC® document: http://www.opengis.net/doc/TS/wmts-simple/1.0/

Internal reference number of this OGC® document: 13-082r2

Version: 1.0

Category: OGC® Standard

Editor: Joan Masó

OGC® Web Map Tile Service (WMTS)

Simple Profile

Copyright notice

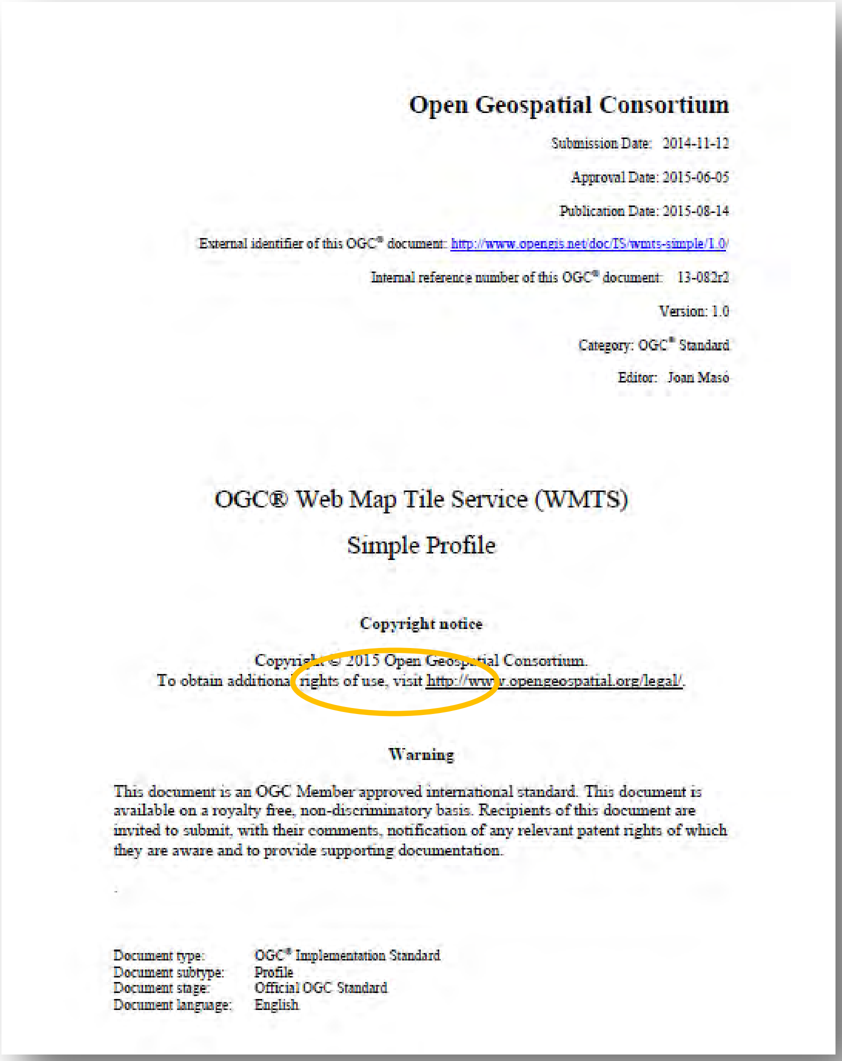
Copyright © 2015 Open Geospatial Consortium.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Implementation Standard
Document subtype:	Profile
Document stage:	Official OGC Standard
Document language:	English

- WMTS-SP defines two possible map projections:
  - Web Mercator
  - Geographic WGS84
- And two possible tiling schemas:
  - GoogleMapsCompatible Tilematrixset
  - WorldCRS84Quad
- Most of the requests that WMTS services receive are for GoogleMapsCompatible Tilematrixset so producers of information are "obliged" to support it



Open Geospatial Consortium

Submission Date: 2014-11-12

Approval Date: 2015-06-05

Publication Date: 2015-08-14

External identifier of this OGC® document: http://www.opengis.net/doc/TS/wmts-simple/1.0/

Internal reference number of this OGC® document: 13-082r2

Version: 1.0

Category: OGC® Standard

Editor: Joan Masó

OGC® Web Map Tile Service (WMTS)

Simple Profile

Copyright notice

Copyright © 2015 Open Geospatial Consortium.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Implementation Standard
Document subtype:	Profile
Document stage:	Official OGC Standard
Document language:	English

Geographic projection

rectangular buildings look rhomboidal

*60° North **Geographic** projection*

roundabouts look ellipsoidal

*42° North **Geographic** projection*

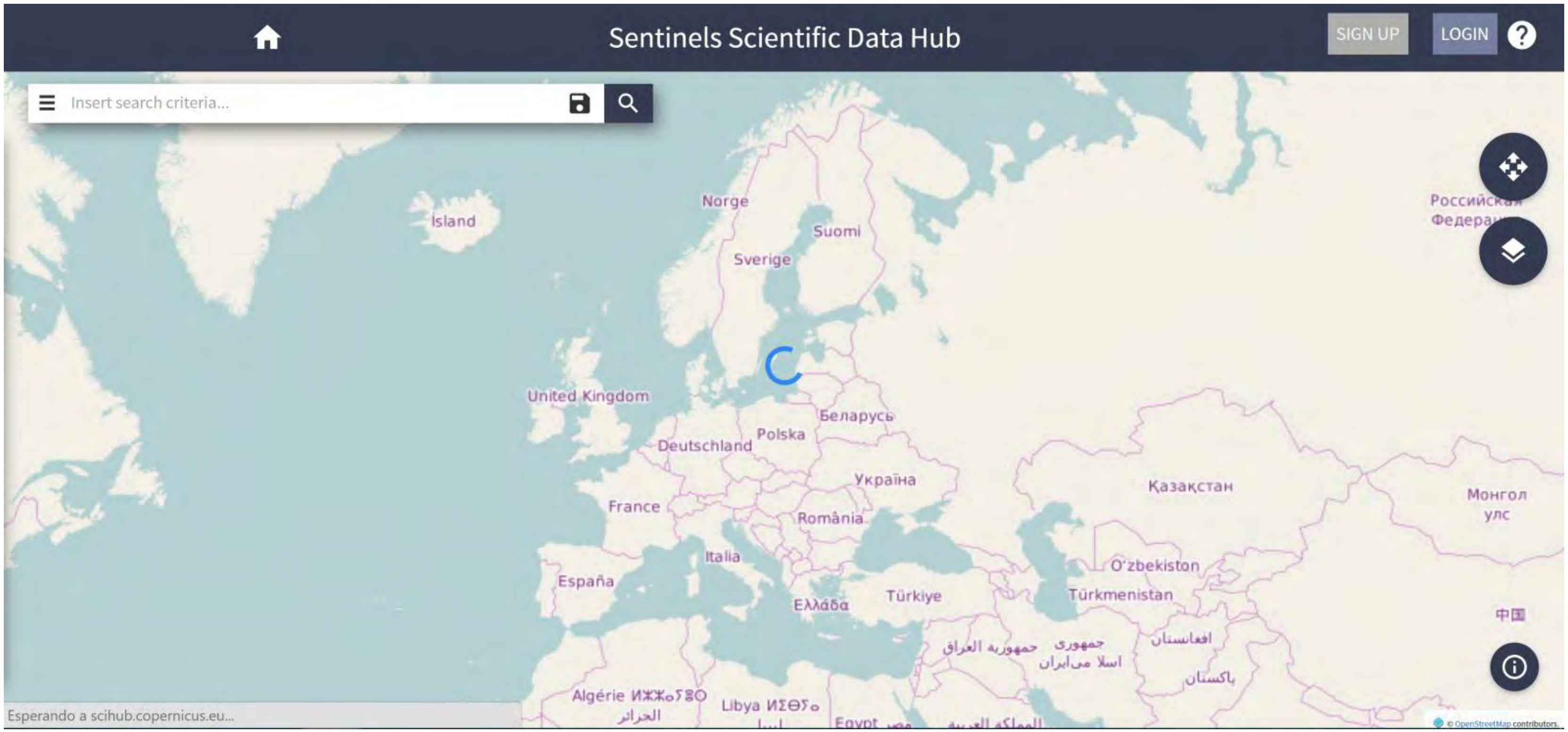
Mercator Projection

rectangular buildings look rectangular

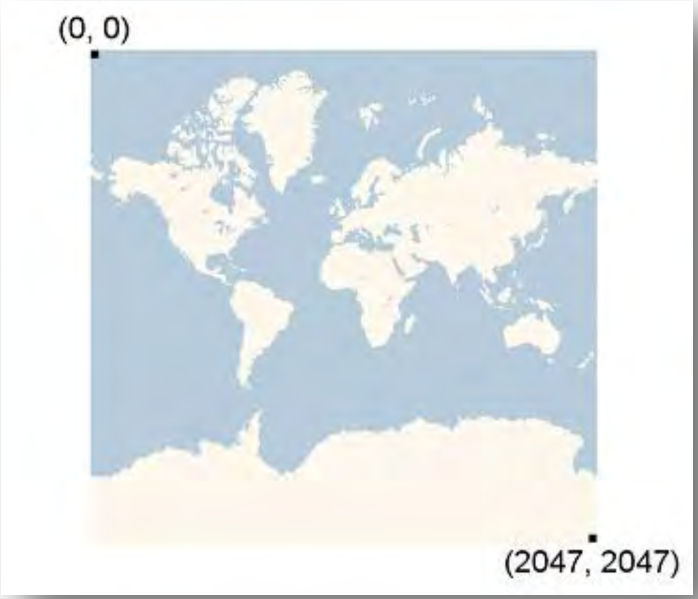
*60° North **Mercator** projection*

roundabouts look circular

*42° North **Mercator** projection*



- Even the web of COPERNICUS Program uses Web Mercator instead of INSPIRE official projections



- Web Mercator does another interesting thing: as it is necessary to "cut at certain latitudes to avoid infinite coordinates, it cuts at the exact latitudes that produce a square:

$\pm$85.05112878º

- Web Mercator recibes many different names: "Spherical Mercator", "Mercator with auxiliary sphere", "Google´s projection",…(and even several different EPSG codes)

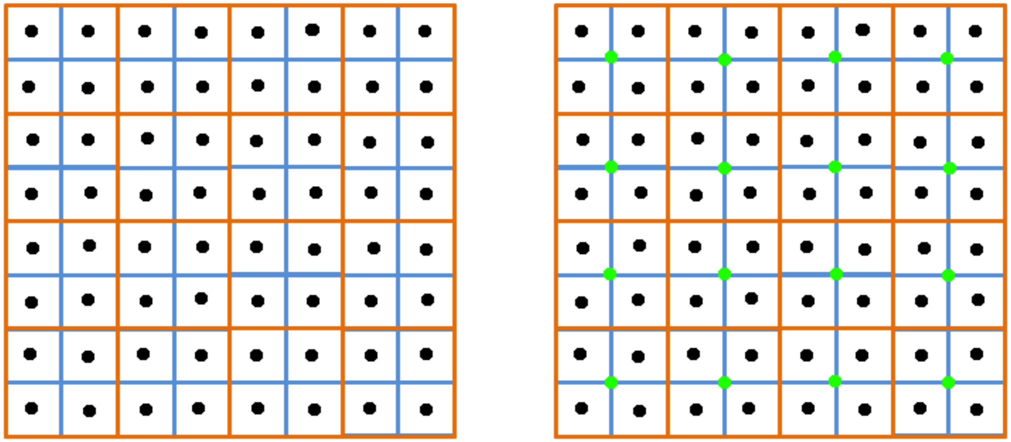
9. Application to Digital Elevation Models

Digital Elevation Models and Orthoimages are mutually complementary for several reasons:

- DEMS are needed to orthorectify images once they have been captured by the sensor
- DEMS are also needed to perform some radiometric corrections such as topographic shadows corrections
- Orthoimages and DEMS can be combined to generate 3D (or 2.5 D) modeling
- etc…

For these reasons, it is very important to maximize interoperability between both kinds of datasets. To attain this interoperability, it is imperative that they share a common grid and tiling schema.

Requirements for DEMs



In black: centers of blue pixels (LOD=n). In green: centers of red pixels (LOD=n-1)

1) We must use the same map projection and the same Tiling Schema as for orthoimages
2) Sampling distances must be the same as those in the list of GSD (LOD)
3) Height measures must correspond to the centers of the image pixels of the same GSD

10. The problem of irrational pixel sizes and irrational corner coordinates

- Pixel sizes (meters/pixel) in Table are not integer, as we are used to

- Instead, they are irrational numbers, with an infinite number of decimals

- The same problem happens with the corner coordinates (x,y)

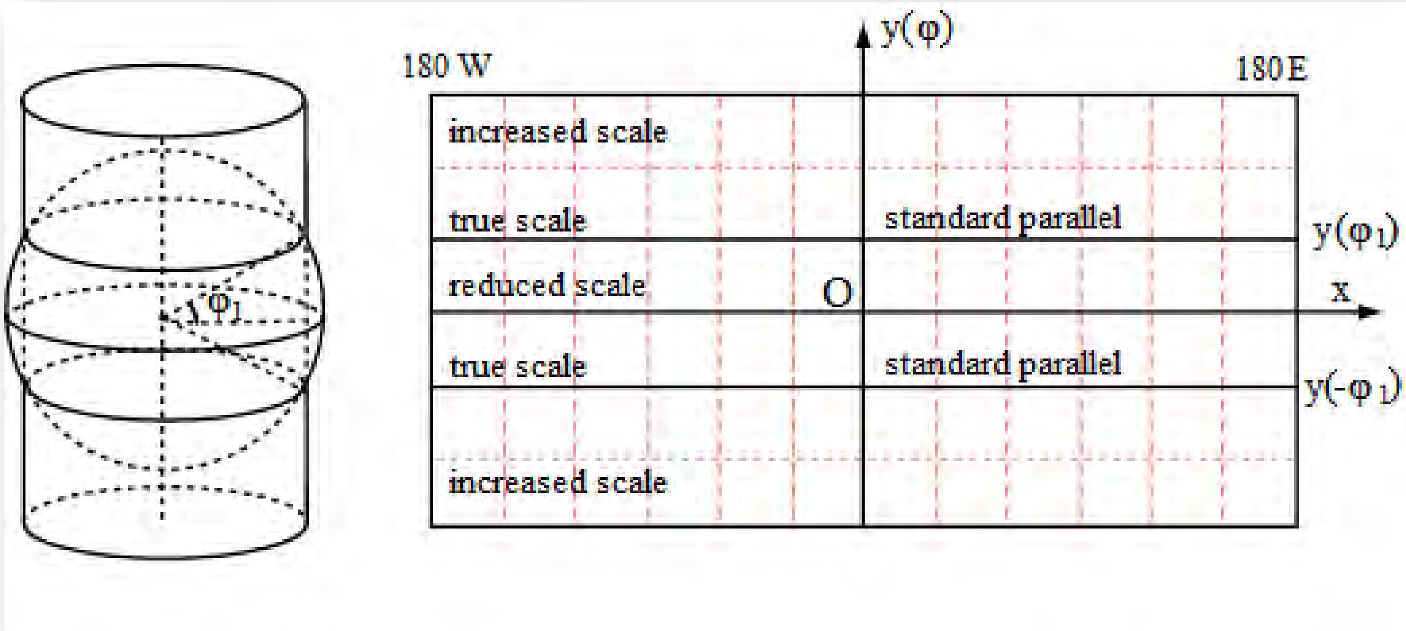
- The operations with real number always have "rounding errors"

- These errors may accumulate when processing a large number of pixels, thus becoming noticeable errors in the form of visual "artifacts" such as "moiré", missing lines, and other problems

- Also, irrational numbers are a "nightmare" for human operators

Level of Detail (LOD)	World Width and Height (pixels)	Pixel sizes (meters / pixel) in equator	Map Scale at 96 dpi
0	256	156.543,033928041	591.658.710,9091310
1	512	78.271,516964021	295.829.355,4545660
2	1.024	39.135,758482010	147.914.677,7272830
3	2.048	19.567,879241005	73.957.338,8636414
4	4.096	9.783,939620503	36.978.669,4318207
5	8.192	4.891,969810251	18.489.334,7159103
6	16.384	2.445,984905126	9.244.667,3579552
7	32.768	1.222,992452563	4.622.333,6789776
8	65.536	611,496226281	2.311.166,8394888
9	131.072	305,748113141	1.155.583,4197444
10	262.144	152,874056570	577.791,7098722
11	524.288	76,437028285	288.895,8549361
12	1.048.576	38,218514143	144.447,9274681
13	2.097.152	19,109257071	72.223,9637340
14	4.194.304	9,554628536	36.111,9818670
15	8.388.608	4,777314268	18.055,9909335
16	16.777.216	2,388657134	9.027,9954668
17	**33.554.432**	**1,194328567**	**4.513,9977334**
18	67.108.864	0,597164283	2.256,9988667
19	134.217.728	0,298582142	1.128,4994333
20	268.435.456	0,149291071	564,2497167
21	536.870.912	0,074645535	282,1248583
22	1.073.741.824	0,037322768	141,0624292
23	2.147.483.648	0,018661384	70,5312146
24	4.294.967.296	0,009330692	35,2656073
25	8.589.934.592	0,004665346	17,6328036
26	17.179.869.184	0,002332673	8,8164018
27	34.359.738.368	0,001166336	4,4082009

Solution: secant Mercator



Secant Mercator projection
(https://en.wikipedia.org/wiki/Mercator_projection)

How to obtain integer pixel sizes, and integer corner coordinates

- If we use a secant Mercator projection, we can calculate the latitude that produces integer pixel sizes in the table of LODs
- For Web Mercator WGS84, the latitude that produces integer pixel sizes happens to be 33.14489729º
- So we should use a "Secant Mercator" projection with two standard parallels at 33.14489729º North and South
- We will call it "Integer Web Mercator"

Level of Detail (LOD)	World Width and Height (pixels)	Pixel sizes (meters / pixel) in equator	Map Scale at 96 dpi	Pixel sizes (meters / pixel) at latitude 33,144897	Map Scales (254 dpi screen) at latitude 33,14489729
0	256	156.543,033928041	591.658.710,9091310	131072,000	1310720000,000
1	512	78.271,516964021	295.829.355,4545660	65536,000	655360000,000
2	1.024	39.135,758482010	147.914.677,7272830	32768,000	327680000,000
3	2.048	19.567,879241005	73.957.338,8636414	16384,000	163840000,000
4	4.096	9.783,939620503	36.978.669,4318207	8192,000	81920000,000
5	8.192	4.891,969810251	18.489.334,7159103	4096,000	40960000,000
6	16.384	2.445,984905126	9.244.667,3579552	2048,000	20480000,000
7	32.768	1.222,992452563	4.622.333,6789776	1024,000	10240000,000
8	65.536	611,496226281	2.311.166,8394888	512,000	5120000,000
9	131.072	305,748113141	1.155.583,4197444	256,000	2560000,000
10	262.144	152,874056570	577.791,7098722	128,000	1280000,000
11	524.288	76,437028285	288.895,8549361	64,000	640000,000
12	1.048.576	38,218514143	144.447,9274681	32,000	320000,000
13	2.097.152	19,109257071	72.223,9637340	16,000	160000,000
14	4.194.304	9,554628536	36.111,9818670	8,000	80000,000
15	8.388.608	4,777314268	18.055,9909335	4,000	40000,000
16	16.777.216	2,388657134	9.027,9954668	2,000	20000,000
17	**33.554.432**	**1,194328567**	**4.513,9977334**	**1,000**	**10000,000**
18	67.108.864	0,597164283	2.256,9988667	0,500	5000,000
19	134.217.728	0,298582142	1.128,4994333	0,250	2500,000
20	268.435.456	0,149291071	564,2497167	0,125	1250,000
21	536.870.912	0,074645535	282,1248583	0,063	625,000
22	1.073.741.824	0,037322768	141,0624292	0,031	312,500
23	2.147.483.648	0,018661384	70,5312146	0,016	156,250
24	4.294.967.296	0,009330692	35,2656073	0,008	78,125
25	8.589.934.592	0,004665346	17,6328036	0,004	39,063
26	17.179.869.184	0,002332673	8,8164018	0,002	19,531
27	34.359.738.368	0,001166336	4,4082009	0,001	9,766

How to obtain integer map scales

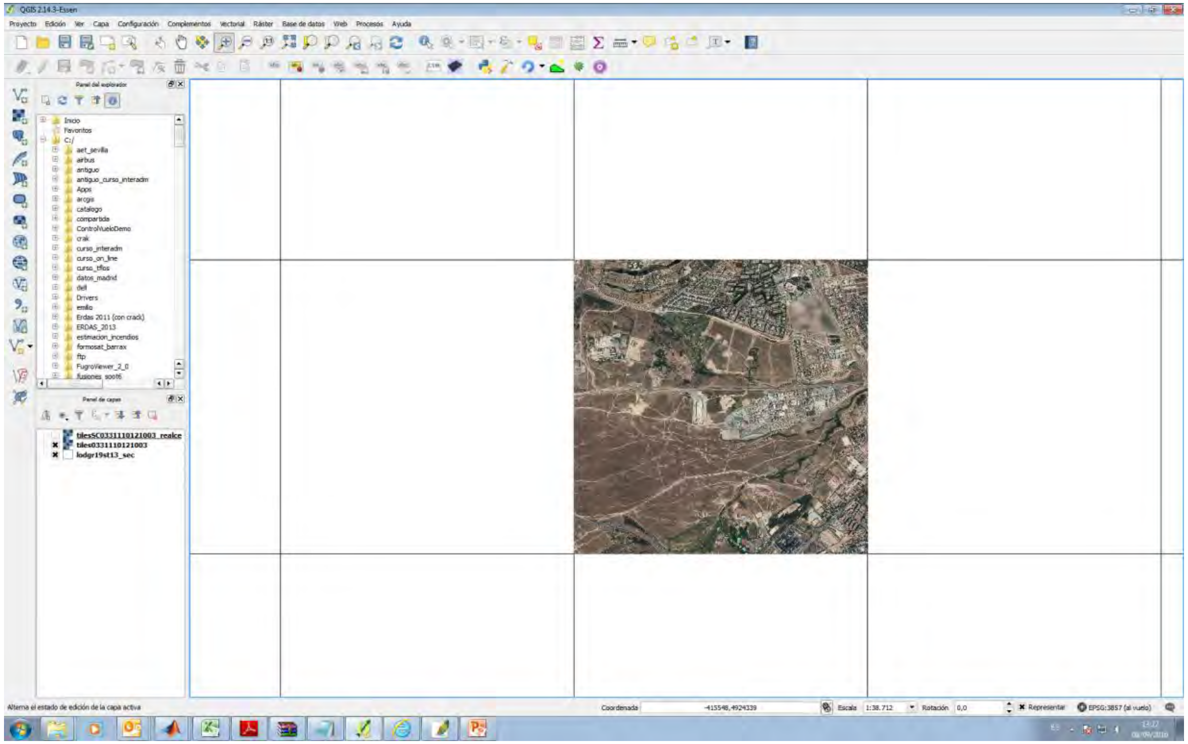
- Real scale on a screen depends on the resolution (pixels per inch) of each screen. There is no obligation to use the “old monitors” resolution of 96 dpi (dots per inch) as a “reference” screen resolution. This produces “ugly” scale numbers even with “round” pixel sizes.

- Today, there exist a great variety of screen resolutions (up to 400 dpi and more), so we can choose a screen resolution that produces “easy to remember” integer Map Scales. E.g: 10 pixels/mm = 254 ppi.

- If we use this resolution, the LOD list looks much better, as it has integer pixel sizes, and integer map scales

Level of Detail (LOD)	World Width and Height (pixels)	Pixel sizes (meters / pixel) in equator	Map Scale at 96 dpi	Pixel sizes (meters / pixel) at latitude	Map Scales (254 dpi screen) at latitude
				33,144897	33,14489729
0	256	156.543,033928041	591.658.710,9091310	131072,000	1310720000,000
1	512	78.271,516964021	295.829.355,4545660	65536,000	655360000,000
2	1.024	39.135,758482010	147.914.677,7272830	32768,000	327680000,000
3	2.048	19.567,879241005	73.957.338,8636414	16384,000	163840000,000
4	4.096	9.783,939620503	36.978.669,4318207	8192,000	81920000,000
5	8.192	4.891,969810251	18.489.334,7159103	4096,000	40960000,000
6	16.384	2.445,984905126	9.244.667,3579552	2048,000	20480000,000
7	32.768	1.222,992452563	4.622.333,6789776	1024,000	10240000,000
8	65.536	611,496226281	2.311.166,8394888	512,000	5120000,000
9	131.072	305,748113141	1.155.583,4197444	256,000	2560000,000
10	262.144	152,874056570	577.791,7098722	128,000	1280000,000
11	524.288	76,437028285	288.895,8549361	64,000	640000,000
12	1.048.576	38,218514143	144.447,9274681	32,000	320000,000
13	2.097.152	19,109257071	72.223,9637340	16,000	160000,000
14	4.194.304	9,554628536	36.111,9818670	8,000	80000,000
15	8.388.608	4,777314268	18.055,9909335	4,000	40000,000
16	16.777.216	2,388657134	9.027,9954668	2,000	20000,000
17	**33.554.432**	**1,194328567**	**4.513,9977334**	**1,000**	**10000,000**
18	67.108.864	0,597164283	2.256,9988667	0,500	5000,000
19	134.217.728	0,298582142	1.128,4994333	0,250	2500,000
20	268.435.456	0,149291071	564,2497167	0,125	1250,000
21	536.870.912	0,074645535	282,1248583	0,063	625,000
22	1.073.741.824	0,037322768	141,0624292	0,031	312,500
23	2.147.483.648	0,018661384	70,5312146	0,016	156,250
24	4.294.967.296	0,009330692	35,2656073	0,008	78,125
25	8.589.934.592	0,004665346	17,6328036	0,004	39,063
26	17.179.869.184	0,002332673	8,8164018	0,002	19,531
27	34.359.738.368	0,001166336	4,4082009	0,001	9,766

11. Pixel to pixel exact correspondence



- An Orthoimage in Web Mercator and Integer Web Mercator are composed of exactly the same pixels, only changing the georreferencing header (e.g: TFW+.prj)

- This allows us to work internally in integer pixel sizes and coordinates and "publish" the data as "standard" Web Mercator

Coordinates of a tile in:
- Web Mercator
- Integer Web Mercator

wkt_geom lod quadkey tile_x tile_y

- Polygon ((-450061.22299999999813735 5014269.05599999986588955, -445169.25300000002607703 5014269.05599999986588955, -445169.25300000002607703 5009377.08600000012665987, -450061.22299999999813735 5009377.08600000012665987, -450061.22299999999813735 5014269.05599999986588955)) 13 0313332322322 4004 3071

wkt_geom lod quadkey tile_x tile_y

Polygon ((-376832 4198400, -372736 4198400, -372736 4194304, -376832 4194304, -376832 4198400)) 13 0313332322322 4004 3071

12. “SuperTiles”

- 256x256 tiles are way too small to be practical as “production units” (“sheets”) for uncompressed orthos, and even more for the compressed mosaics
- So we propose to use as production units the same footprints of the Tiling Schema, but with a pixel size of other LOD.
- For example: if we have LOD 17 pixel size and use LOD 11 tiles footprints as “production units”, they would be 256x64= 16,384 x 16,384 pixels
- For short we will call these images “SuperTiles”.

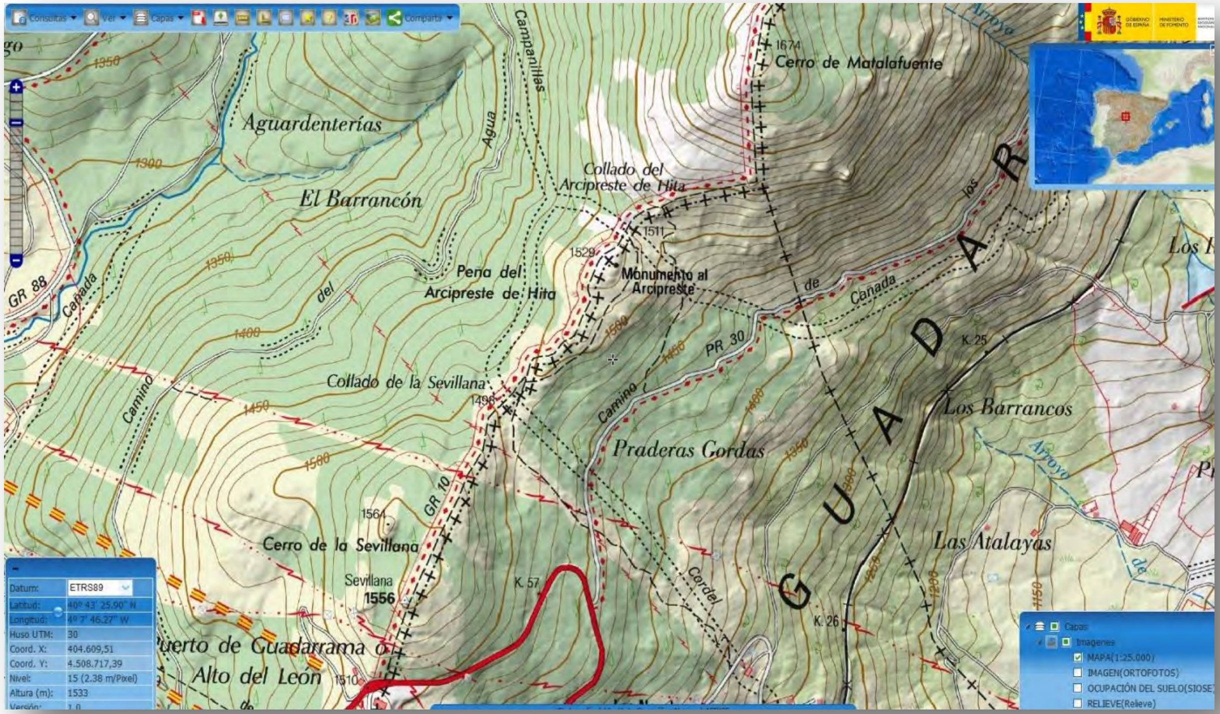
Level of Detail (LOD)	World Width and Height (pixels)	Pixel sizes (meters / pixel) at latitude	Map Scales (254 dpi screen) at latitude
		33,144897	33,14489729
0	256	131072,000	1310720000,000
1	512	65536,000	655360000,000
2	1.024	32768,000	327680000,000
3	2.048	16384,000	163840000,000
4	4.096	8192,000	81920000,000
5	8.192	4096,000	40960000,000
6	16.384	2048,000	20480000,000
7	32.768	1024,000	10240000,000
8	65.536	512,000	5120000,000
9	131.072	256,000	2560000,000
10	262.144	128,000	1280000,000
11	524.288	64,000	640000,000
12	1.048.576	32,000	320000,000
13	2.097.152	16,000	160000,000
14	4.194.304	8,000	80000,000
15	8.388.608	4,000	40000,000
16	16.777.216	2,000	20000,000
17	**33.554.432**	**1,000**	**10000,000**
18	67.108.864	0,500	5000,000
19	134.217.728	0,250	2500,000
20	268.435.456	0,125	1250,000
21	536.870.912	0,063	625,000
22	1.073.741.824	0,031	312,500
23	2.147.483.648	0,016	156,250
24	4.294.967.296	0,008	78,125
25	8.589.934.592	0,004	39,063
26	17.179.869.184	0,002	19,531
27	34.359.738.368	0,001	9,766

13. BigTiles

- Compressed mosaics should be the composition of e.g: 8x8 "SuperTiles"
- For short we will call these "BigTiles"

Level of Detail (LOD)	World Width and Height (pixels)	Pixel sizes (meters / pixel) at latitude	Map Scales (254 dpi screen) at latitude
		33,144897	33,14489729
0	256	131072,000	1310720000,000
1	512	65536,000	655360000,000
2	1.024	32768,000	327680000,000
3	2.048	16384,000	163840000,000
4	4.096	8192,000	81920000,000
5	8.192	4096,000	40960000,000
6	16.384	2048,000	20480000,000
7	32.768	1024,000	10240000,000
$\boxed{8}$	65.536	512,000	5120000,000
9	131.072	256,000	2560000,000
10	262.144	128,000	1280000,000
11	524.288	64,000	640000,000
12	1.048.576	32,000	320000,000
13	2.097.152	16,000	160000,000
14	4.194.304	8,000	80000,000
15	8.388.608	4,000	40000,000
16	16.777.216	2,000	20000,000
$\boxed{\mathbf{17}}$	**33.554.432**	**1,000**	**10000,000**
18	67.108.864	0,500	5000,000
19	134.217.728	0,250	2500,000
20	268.435.456	0,125	1250,000
21	536.870.912	0,063	625,000
22	1.073.741.824	0,031	312,500
23	2.147.483.648	0,016	156,250
24	4.294.967.296	0,008	78,125
25	8.589.934.592	0,004	39,063
26	17.179.869.184	0,002	19,531
27	34.359.738.368	0,001	9,766

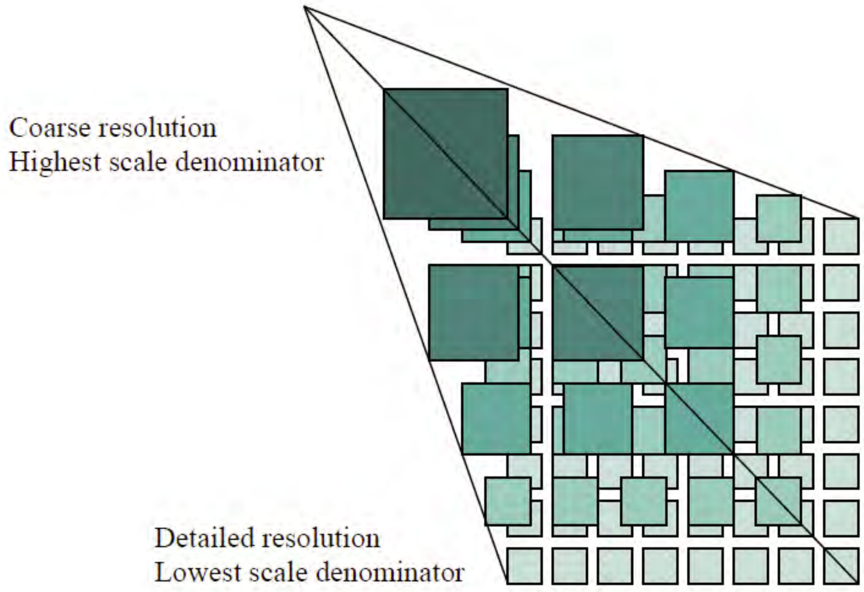
14. Application to raster maps

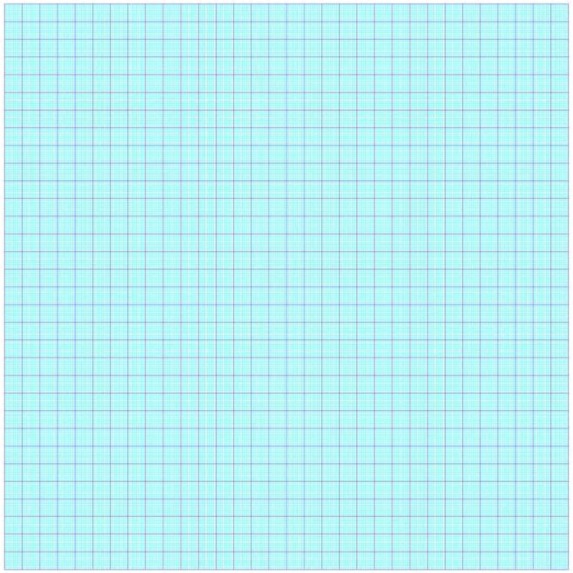


All this reasoning is also applicable to raster maps

15. Problem of the huge number of WMTS tiles

- WMTS services require a huge number of tiles: hundreds of millions of individual “tiny” 256x256 JPEG files must be produced (either pre-cached or “on the fly”) in one or several “projections”
- These tiles are very difficult to manage in current computing environments, because operating systems are not prepared for such a large number of files.

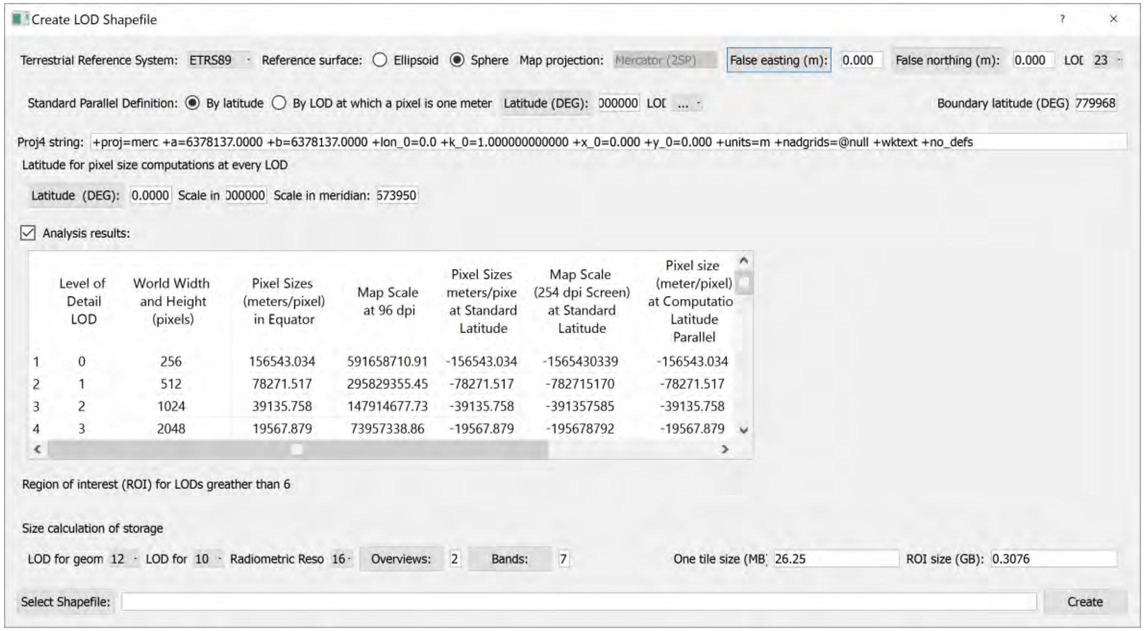


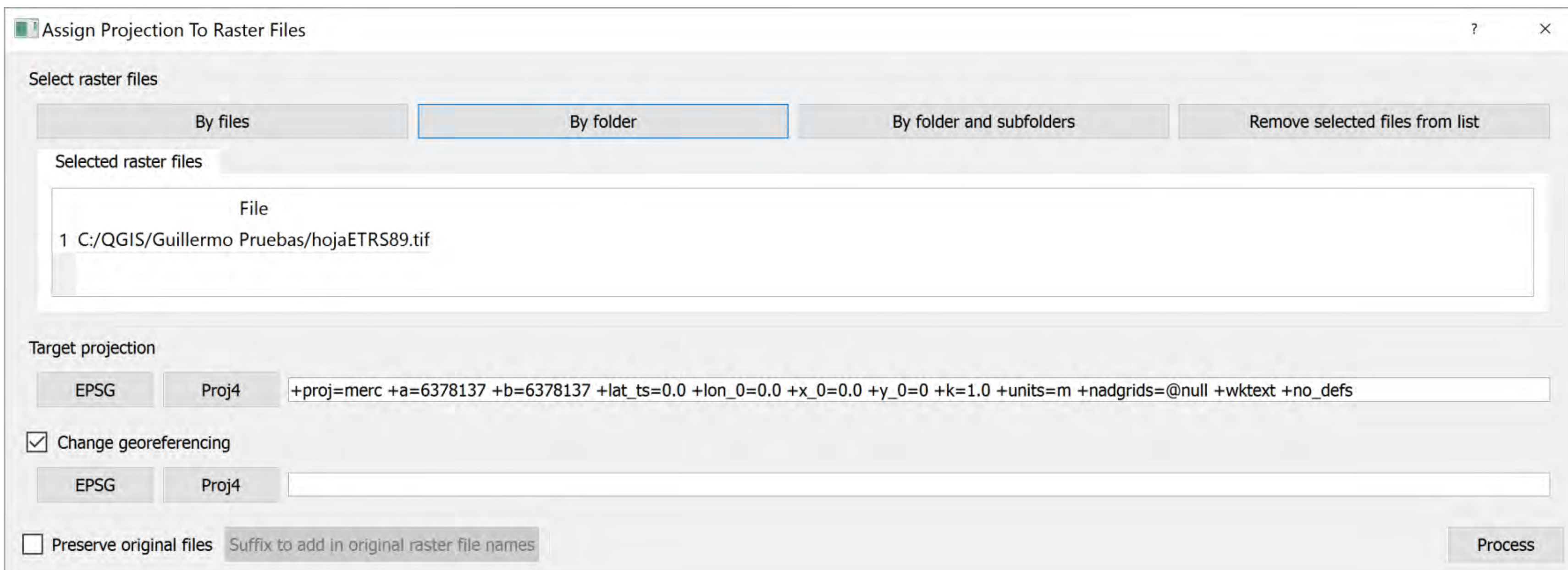


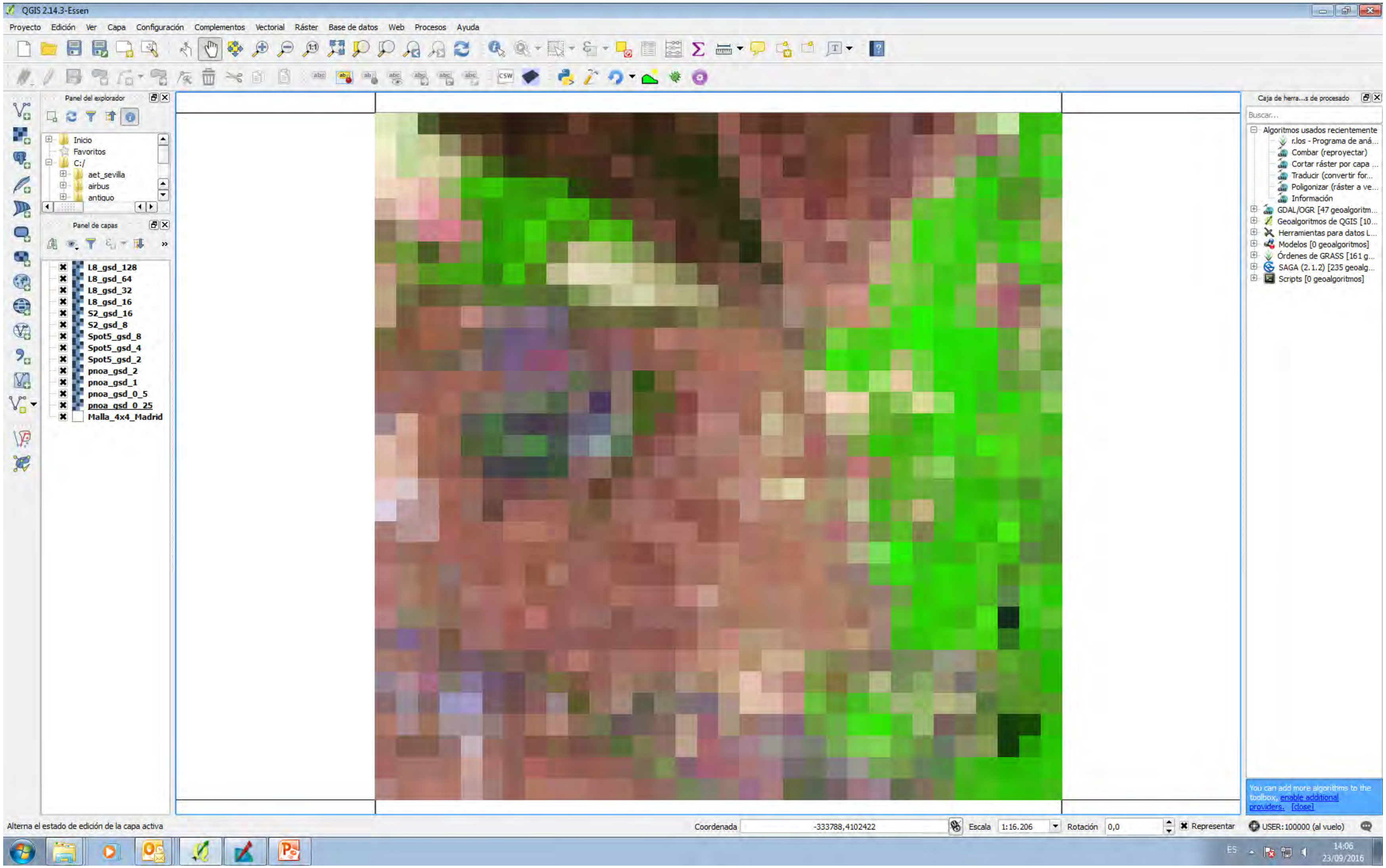
- A solution is to store many of the 256x256 tiles “inside” a bigger file: TiledTIFF (internal file-based tiling) is the best place to store tiles of an WMTS service. We only need to compress them in JPEG.
- If we generate a TiledTIFF with JPEG compression using the footprints and the pixel sizes of WMTS-SP we obtain WMTS tiles ready to be directly sent without the need to decompress and recompress them before the delivery
- This approach has already been implemented by Mapserver opensource project (*http://mapserver.org/es/mapcache/caches.html#geo-tiff-caches*).

16. Nested grid and Integer Web Mercator in the practice

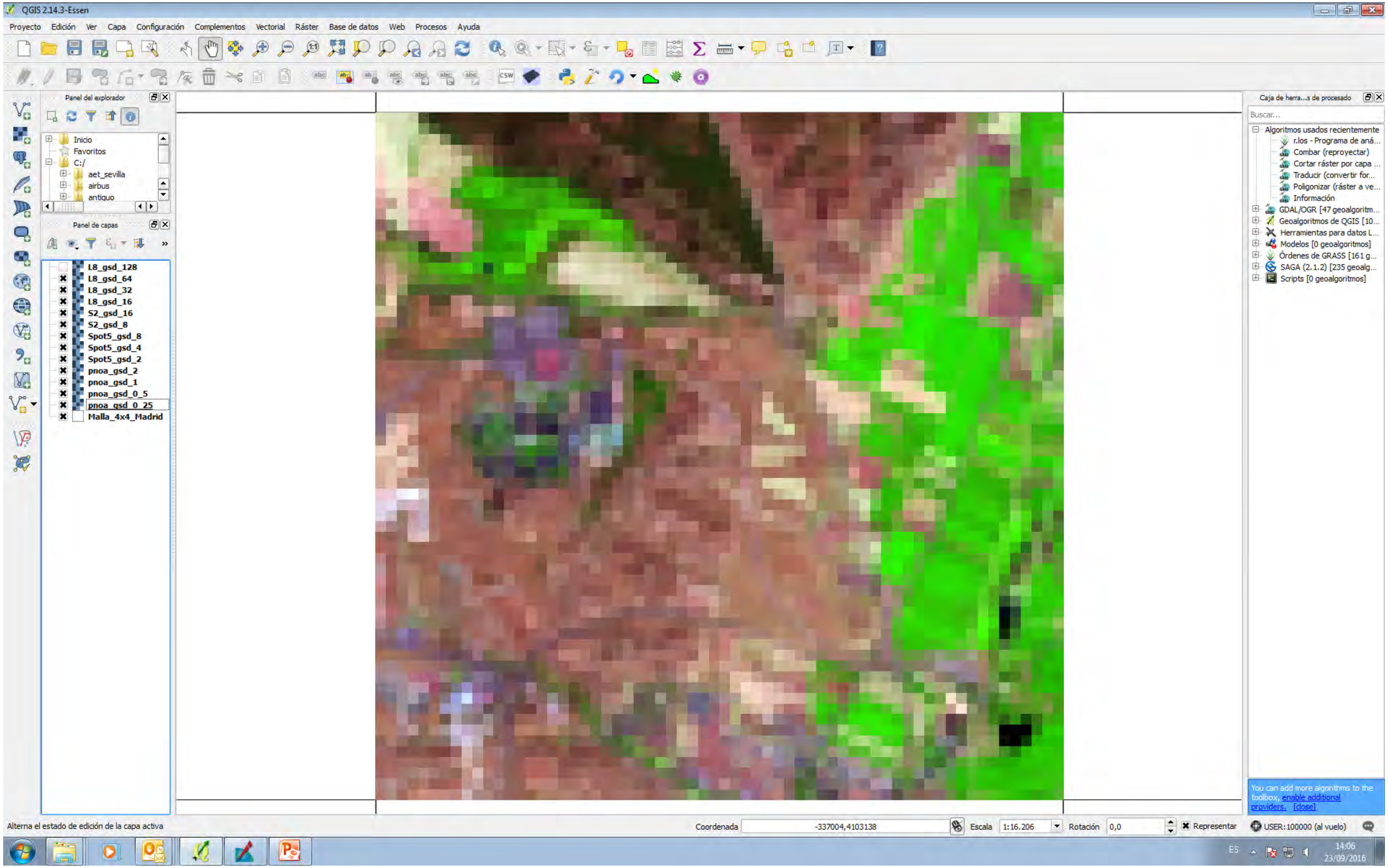
“Nested grid Tools” development



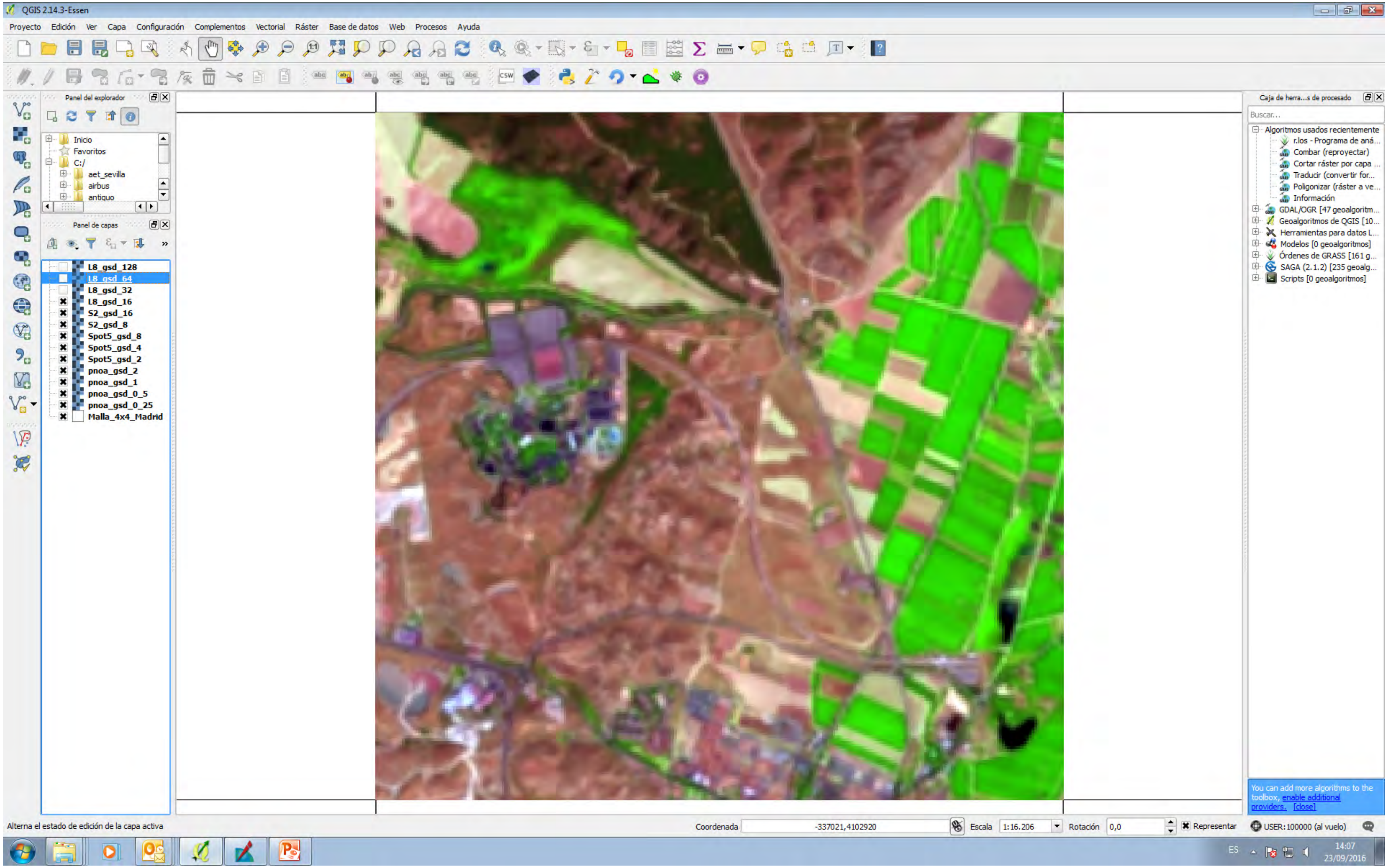




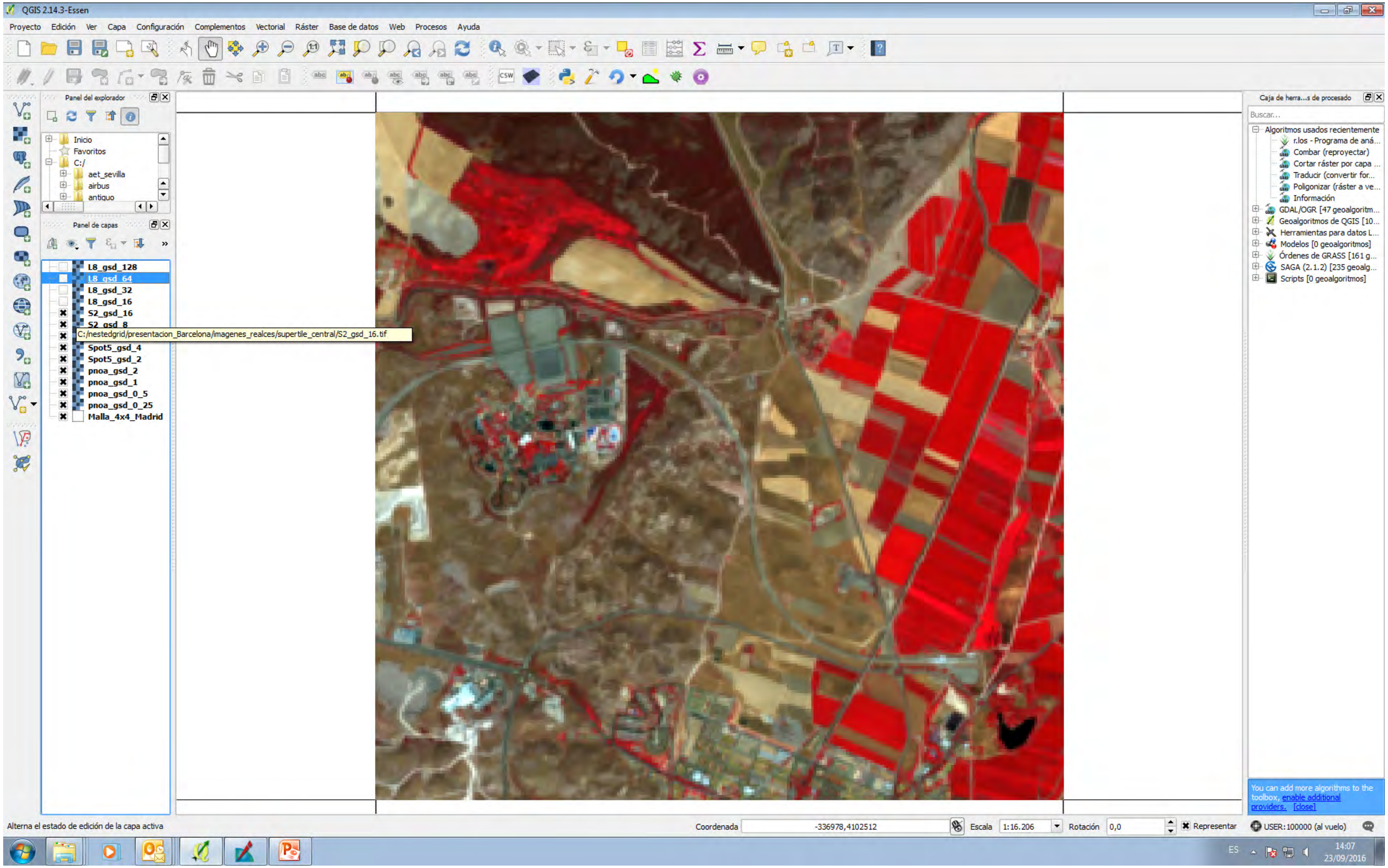
Landsat 8 64m



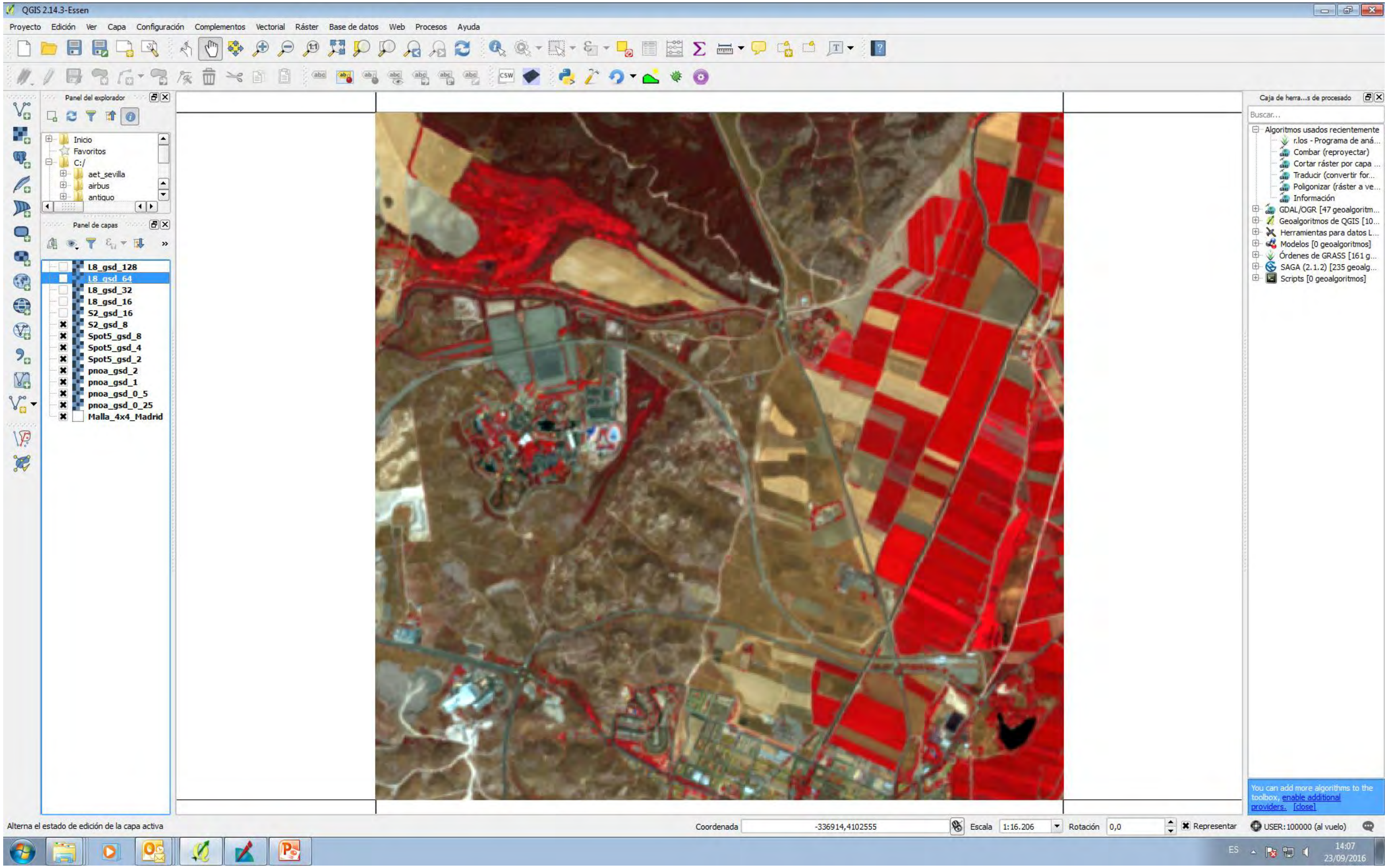
Landsat 8 32m



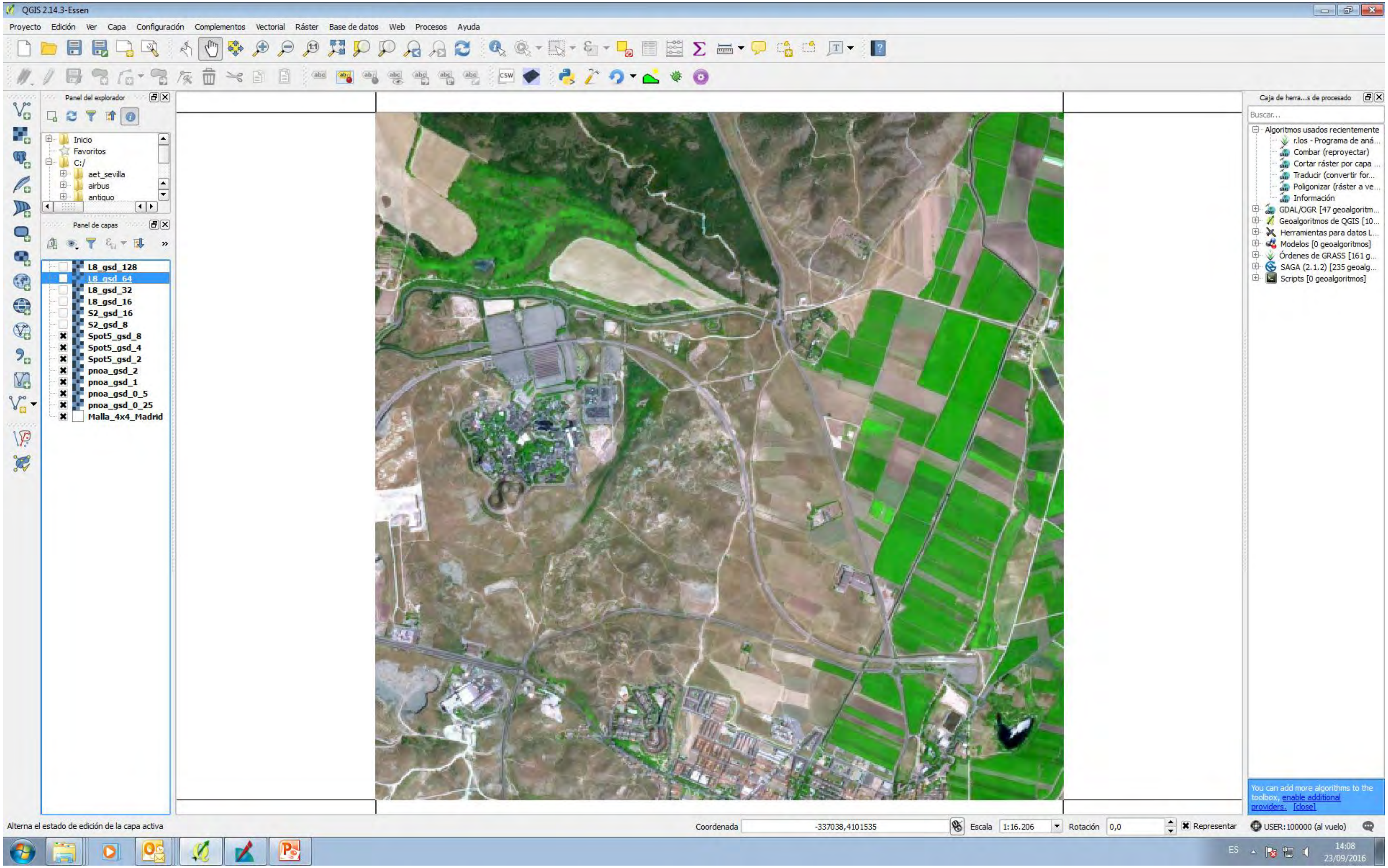
Landsat 8   16m



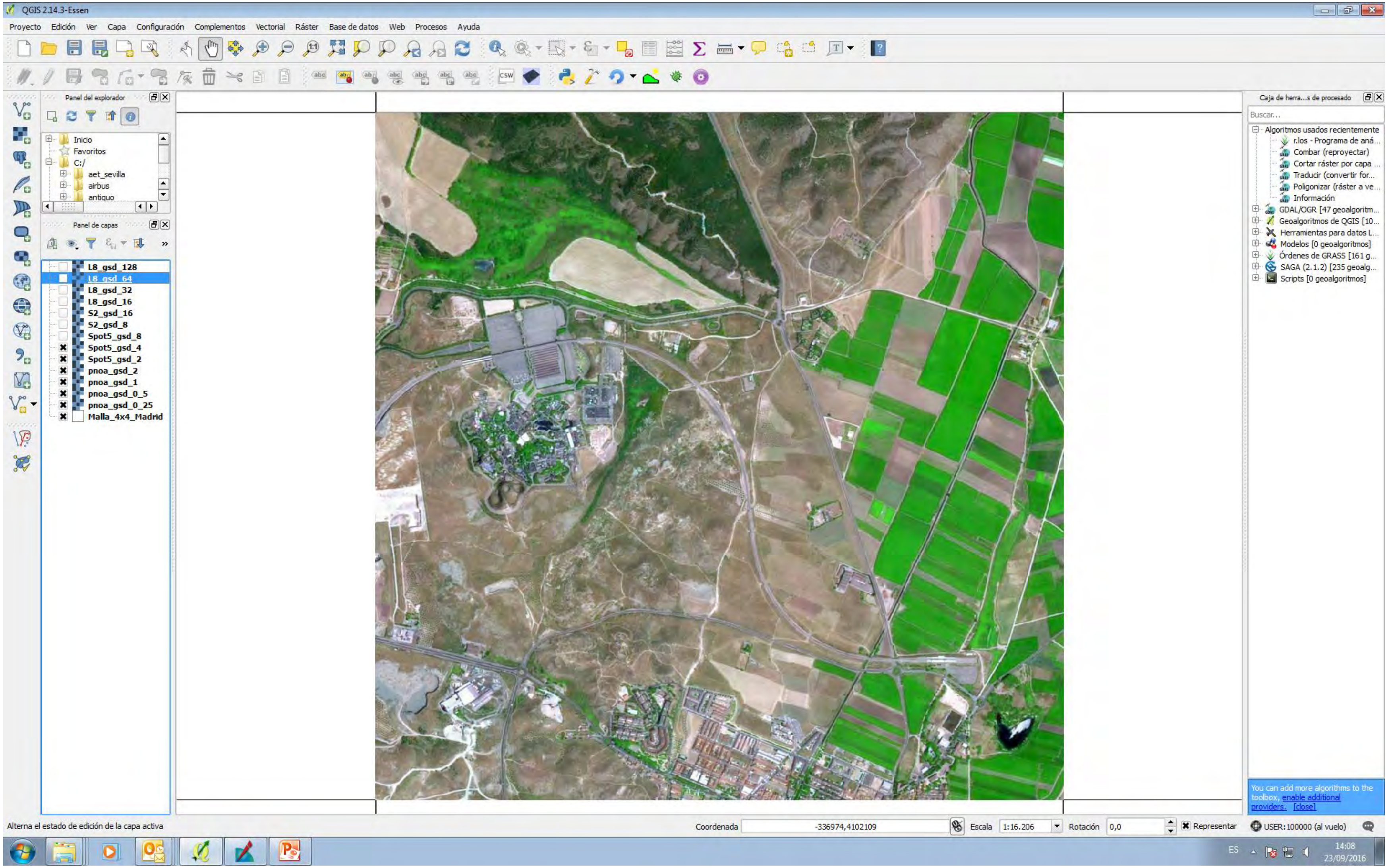
Sentinel 2   16m



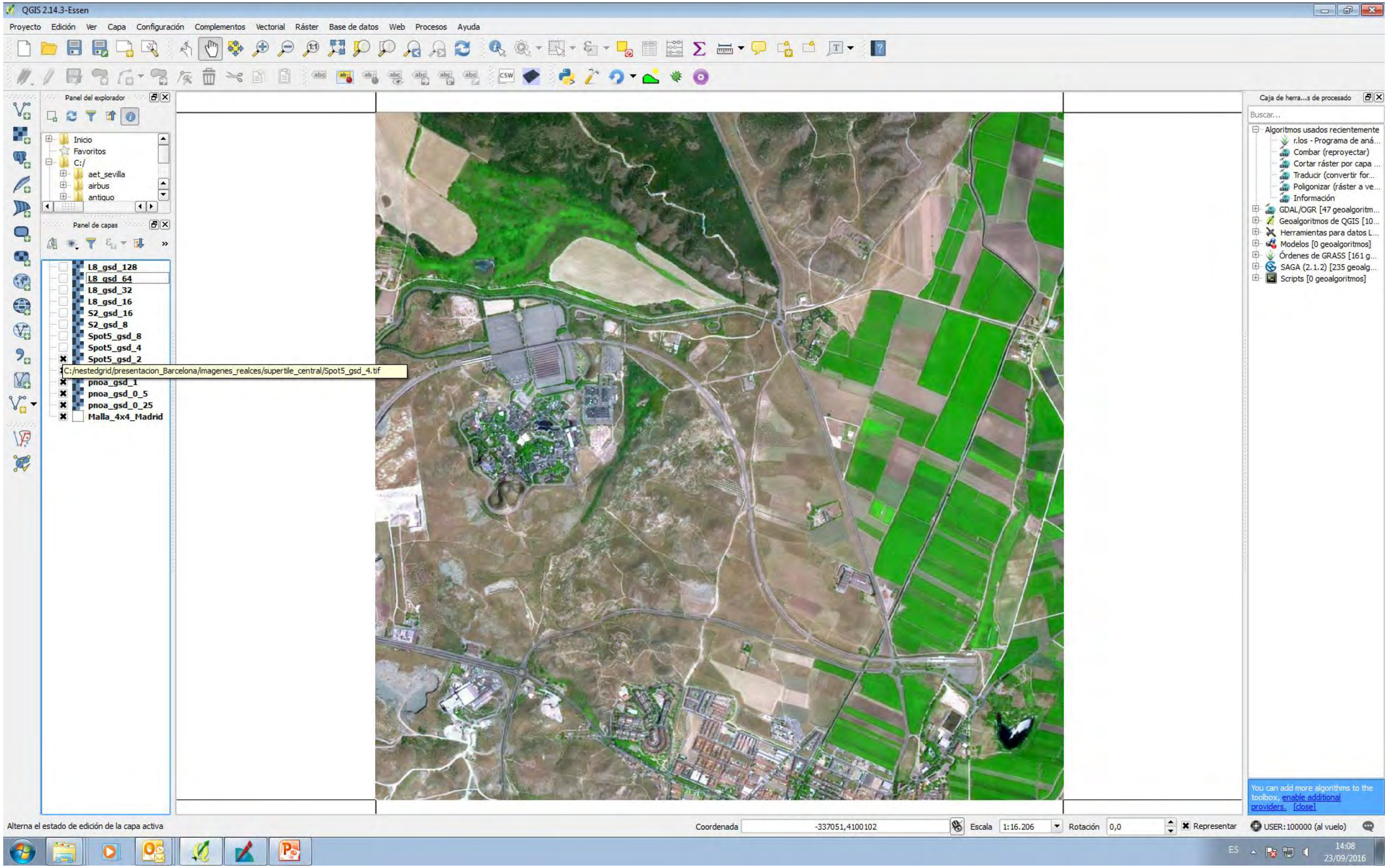
Sentinel 2   8m


QGIS 2.14.3-Essen
Proyecto Edición Ver Capa Configuración Complementos Vectorial Ráster Base de datos Web Procesos Ayuda
Panel del explorador
Inicio
Favoritos
C:/
aet_sevilla
airbus
antiguo
Panel de capas
L8_gsd_128
L8_gsd_64
L8_gsd_32
L8_gsd_16
S2_gsd_16
S2_gsd_8
Spot5_gsd_8
Spot5_gsd_4
Spot5_gsd_2
pnoa_gsd_2
pnoa_gsd_1
pnoa_gsd_0_5
pnoa_gsd_0_25
Malla_4x4_Madrid
Caja de herra...s de procesado
Buscar...
Algoritmos usados recientemente
r.los - Programa de aná...
Combar (reproyectar)
Cortar ráster por capa ...
Traducir (convertir for...
Poligonizar (ráster a ve...
Información
GDAL/OGR [47 geoalgoritm...
Geoalgoritmos de QGIS [10...
Herramientas para datos L...
Modelos [0 geoalgoritmos]
Órdenes de GRASS [161 g...
SAGA (2.1.2) [235 geoalg...
Scripts [0 geoalgoritmos]
You can add more algorithms to the toolbox, enable additional providers. [close]
Alterna el estado de edición de la capa activa
Coordenada -337038,4101535
Escala 1:16.206
Rotación 0,0
Representar
USER:100000 (al vuelo)
ES
14:08
23/09/2016

Spot 5 8m

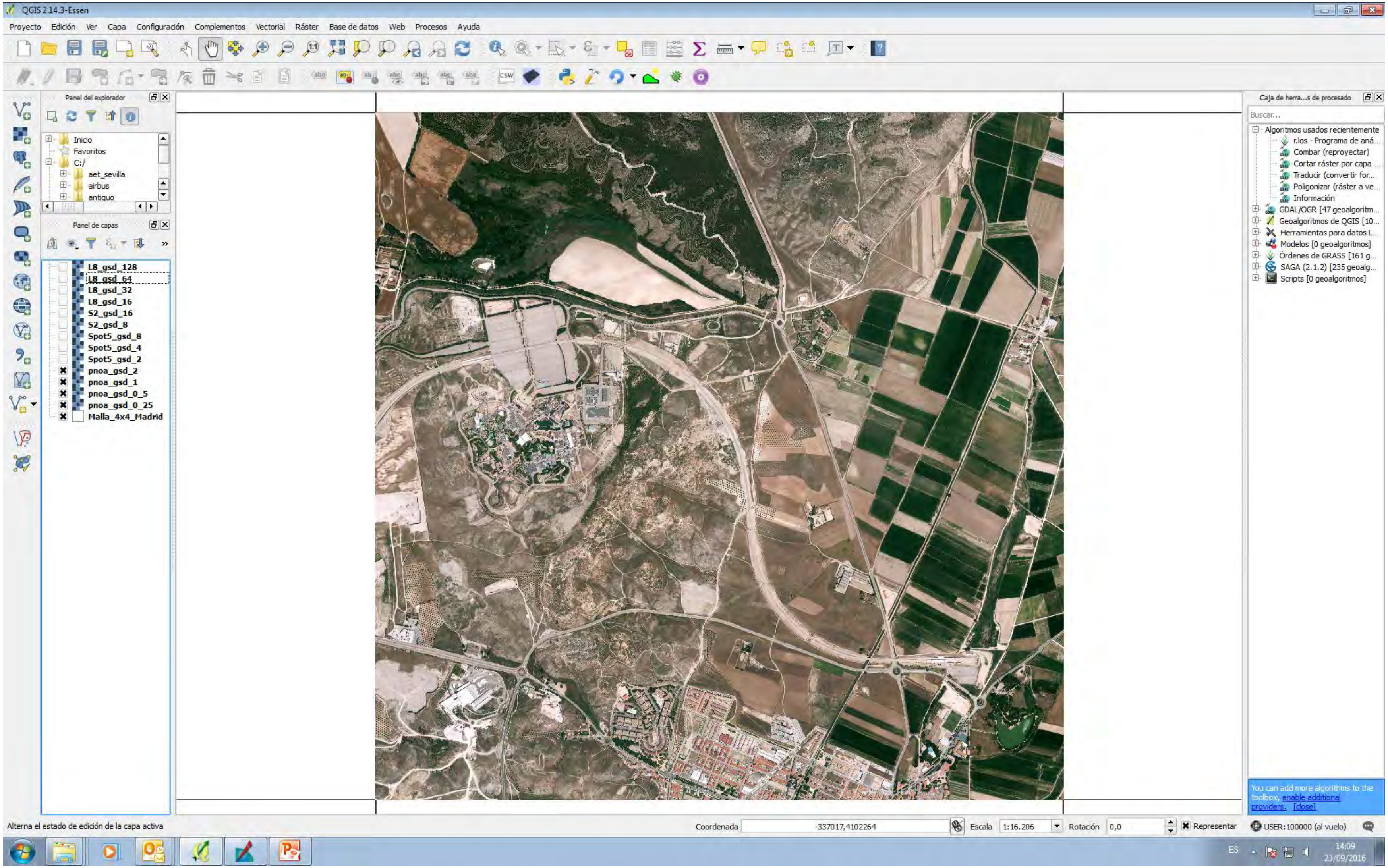


Spot 5   4m

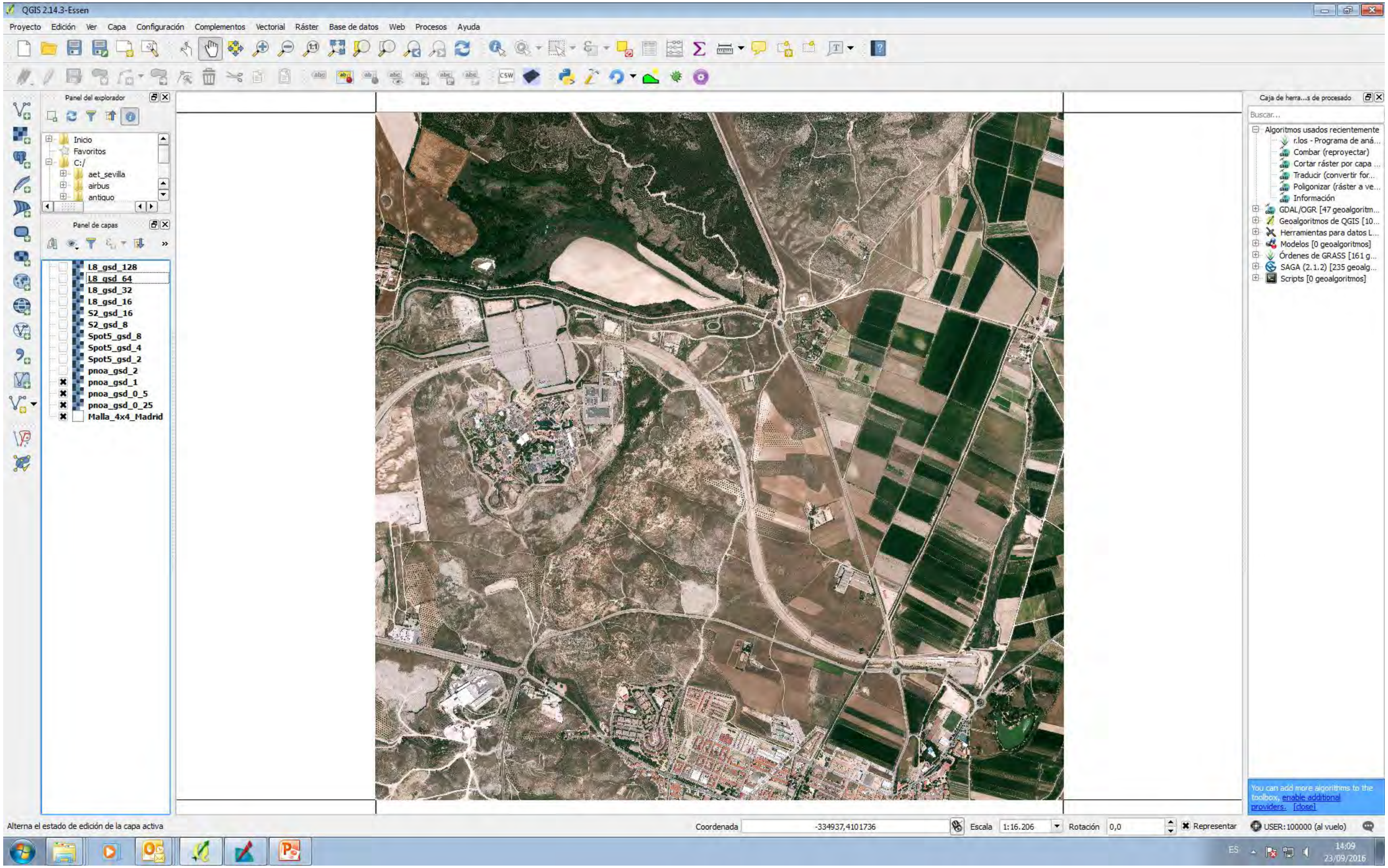


QGIS 2.14.3-Essen
Proyecto Edición Ver Capa Configuración Complementos Vectorial Ráster Base de datos Web Procesos Ayuda
Panel del explorador
Inicio
Favoritos
C:/
aet_sevilla
airbus
antiquo
Panel de capas
L8_gsd_128
L8_gsd_64
L8_gsd_32
L8_gsd_16
S2_gsd_16
S2_gsd_8
Spot5_gsd_8
Spot5_gsd_4
Spot5_gsd_2
C:/nestedgrid/presentacion_Barcelona/imagenes_realces/supertile_central/Spot5_gsd_4.tif
pnoa_gsd_1
pnoa_gsd_0_5
pnoa_gsd_0_25
Malla_4x4_Madrid
Caja de herra...s de procesado
Buscar...
Algoritmos usados recientemente
r.los - Programa de aná...
Combar (reproyectar)
Cortar ráster por capa ...
Traducir (convertir for...
Poligonizar (ráster a ve...
Información
GDAL/OGR [47 geoalgoritm...
Geoalgoritmos de QGIS [10...
Herramientas para datos L...
Modelos [0 geoalgoritmos]
Órdenes de GRASS [161 g...
SAGA (2.1.2) [235 geoalg...
Scripts [0 geoalgoritmos]
You can add more algorithms to the toolbox, enable additional providers. [close]
Alterna el estado de edición de la capa activa
Coordenada -337051,4100102
Escala 1:16.206
Rotación 0,0
Representar
USER:100000 (al vuelo)
ES
14:08
23/09/2016

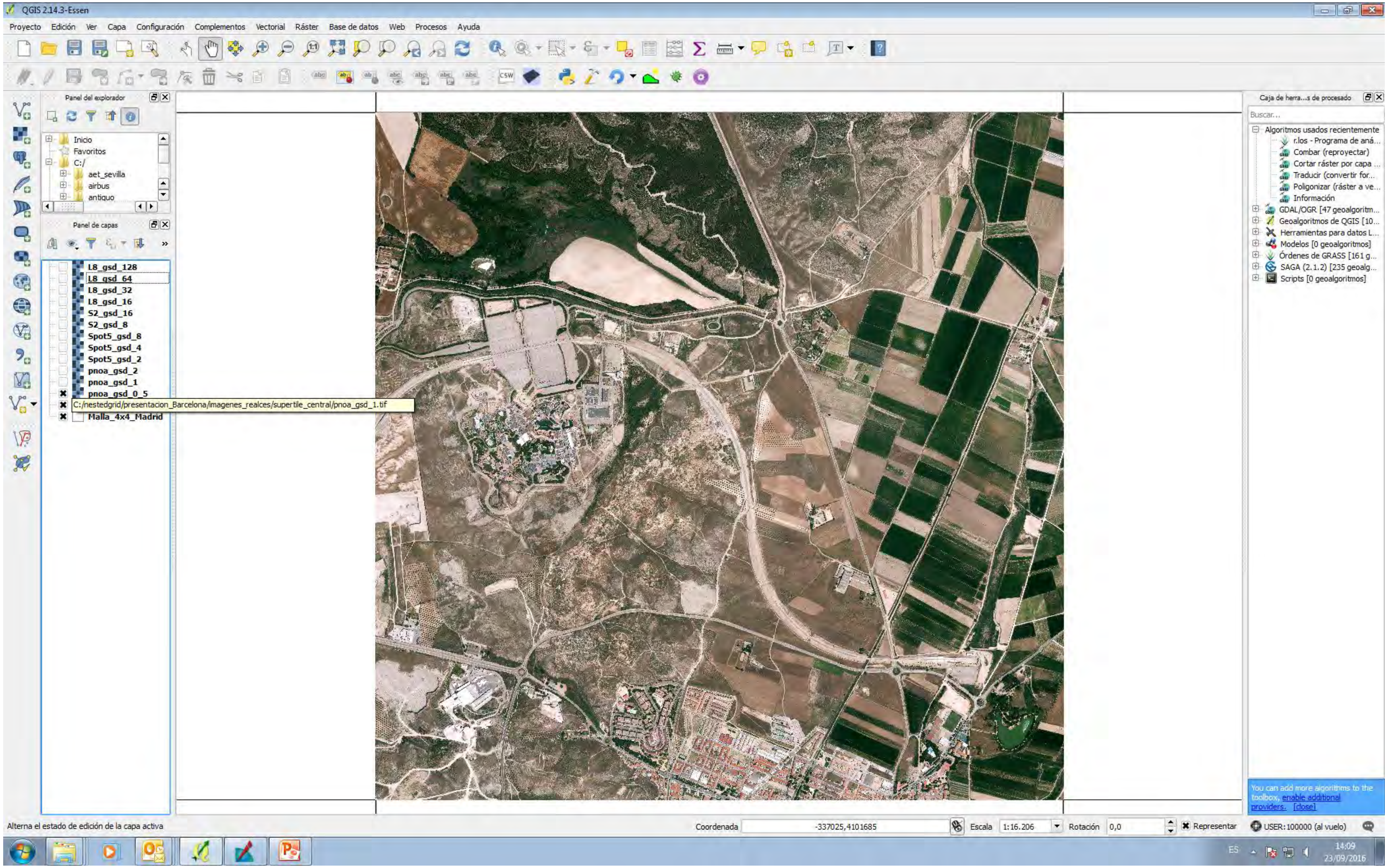
Spot 5 2m



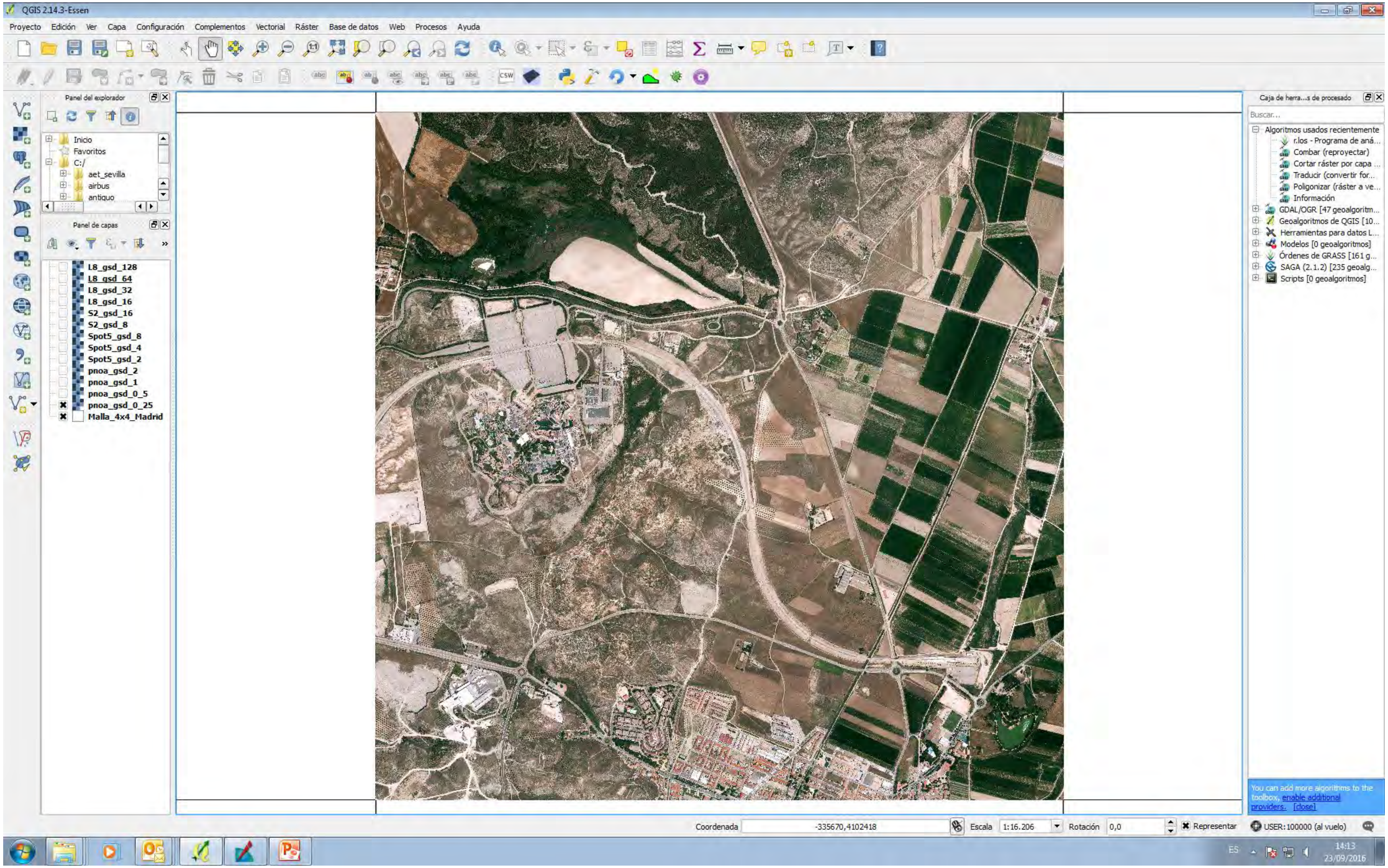
Orthophoto 2m



Orthophoto 1m



Orthophoto 0.5m



Orthophoto 0.25m

16. Proposals for Inspire Data Specifications (I)

- Web Mercator is not among map projections allowed by Inspire

➡ In order to assure interoperability with a high number of tiled web services, the list of recommended Spatial Reference Systems should include **Web Mercator** (and **integer** Web Mercator)

- Data Specifications recommend a common grid (Zoned Geographic Grid) for Orthoimagery and Elevation (Annex D of both DS). However, it does not include a tiling schema

➡ In order to assure interoperability of different datasets and efficient tiled web services, we propose to GoogleMapsCompatible Tilematrixset in the list of recommended common grids for IO and EL

Proposals for Inspire Data Specifications (II)

- "TiledTIFF" and JPEG compressed TIFF are not accepted in Inspire DS

In order to allow this efficient production -> publication workflow, Tiled and JPEG compressed TIFFs should be accepted.

"BigTIFF" and "Pyramidal TIFF" should also be accepted in order to ease this workflow

- Recommend the use of "SuperTiles" as production units for uncompressed orthoimages and Elevations

- Recommend the use of "BigTiles" for compressed orthoimages

Discussion document

- Document describing the proposal uploaded to the INSPIRE Thematic Cluster collaboration platform:

  https://themes.jrc.ec.europa.eu/file/view/76196/2015-11-29-a-nested-grid-for-inspire-ortoimagesdocx

- Discussion topic:

  https://themes.jrc.ec.europa.eu/discussion/view/10935/usability-of-the-zoned-geographic-grid-grid-etrs89-grs80

- **Look forward for receiving your feedback!**