

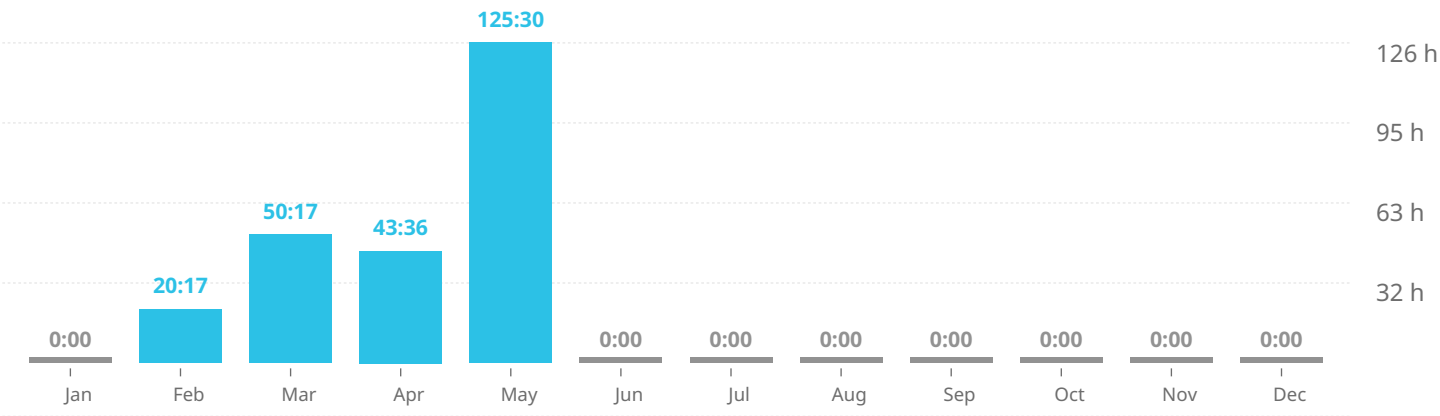
# Summary report



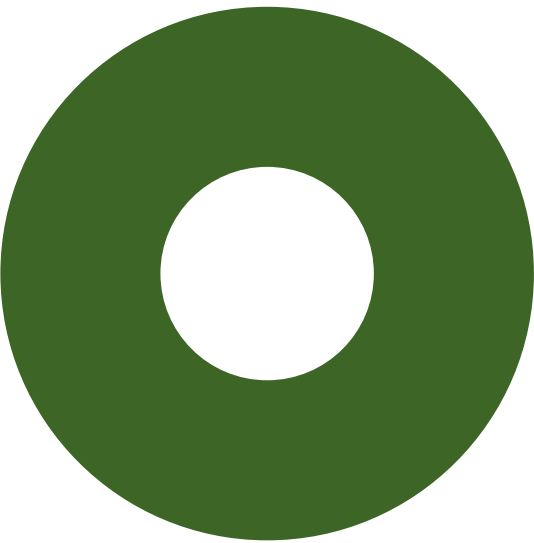
2016-01-01 - 2016-12-31


Total 239 h 41 min

Crileaech, Miguel Angel Cifredo, Victor selected as users  
sos-2016-02 selected as projects






Projects



 sos-2016-02 239:41:21

Time entries



 #110 Preparing D04 for OLDERS. 50:00:00  
 #107 population API: use ... 25:10:00  
 Other 164:31:21

Projects / Time entries	Duration
<b>sos-2016-02</b>	<b>239:41:21</b>
#100 Create Proxy to Group 01	2:41:00
#104 check that all previous deliverables still work + some fixes	3:06:00
#105 Video comentando todas las funcionalidades y aspectos del entregable D04	2:39:00
#106 population API: use governify	2:14:00
#107 population API: use and integrate other APIs	25:10:00
#10 Discuss stuff to do for this week	0:15:00
#110 Preparing D04 for OLDERS.	50:00:00
#11 create repository con my pc.	0:04:58
#12 Create package.json	0:05:43
#13 Edit description of Opera Group	0:10:23
#14 installed node js and sublime	0:08:45
#15 check backlog, try to add Jose M Garcia + email him about problems	0:18:00
#16 Explain some git stuff to Cristina	0:07:00
#19 Create /about/population page	0:43:00
#20 Heroku. Install aplicacion.	0:09:07
#21 2. Heroku. Prepare the app	0:15:18
#22 3. Heroku. Deploy the app.	4:40:23
#23 Find how to share deploy privileges with the team on Heroku	0:32:00
#25 Heroku: Fix problem with repo no synched with GitHub	0:25:00
#29 Fix conflict mess + see description for more	1:57:40
#2 Piazza: Respuesta + una pregunta	0:35:30
#30 Create static folder in Heroku	2:24:14
#31 Clean static pages routing	1:24:00
#37 Decide Git workflow with team	0:25:00
#3 Comprensión de los sistemas.	2:07:19
#3 rename GitHub org + rename Toggl org + Leer respuesta en piazza	0:40:00
#40 Modificar el enrutamiento de las páginas según pedían en la entrega del día 09/03/2016	0:00:14
#44 Creating CRUD for FeedBak 04	4:44:56
#46 Add config file for ESLint, which specifies that we use ES6 + Minor fixes	0:24:00
#47 Implement example REST API	1:22:00
#47 Implement example REST API + handle merge issues with Cristina	4:20:00
#48 Modificar tipo de dato retorno	0:48:00
#4 help Cristina create main team repo + begin filling timetracking and create related issues	0:42:00
#56 Implementar API REST para "population"	8:40:00
#59 population API: implement token auth + investigate dependency issue	3:48:00
#5 Config all systems on my own PC.	3:39:36

#62 API: population: Implement 3 layers routing	1:02:00
#70 REST client: check that it meets requirements	1:03:00
#71 REST client: pretty print	0:54:00
#72 population: table view	1:50:00
#75 population: mejorar la apariencia de la tabla de datos	4:51:00
#78 population: make REST client prettier	1:20:00
#7 manage mess and confusion with various togggl workspaces and the GitHub organisation	0:08:00
#80 population GUI: server side search	1:20:00
#81 population GUI: create, update, destroy	7:52:00
#83 population GUI: Especificar una apikey + #85 population GUI: show errors	1:08:00
#84 population GUI: pagination	1:31:00
#86 grupo: página inicial (/) + check specs are all implemented	0:43:00
#88 population GUI: fix client side search + other small fixes	0:14:00
#8 read piazza + make teachers admin on GitHub org + check what remains to be done in backlog	0:30:00
#90 population GUI: search minimun population	0:38:00
#91 population: pagina de visualización gráfica	6:36:00
#97 proxy to other API	0:58:00
#98 population API: add CORS headers	0:33:00
#9 sort out all GitHub notifications + clean some of the first issues open + check teachers are in togggl	0:14:00
Analyse Backlog + read Piazza	0:26:00
Analyse spec for next deliverable	0:15:00
Ask for new specifications to continue to work	0:10:00
Brief Cristina about the table	0:35:00
Catch up discussions + rant about having worked for nothing	0:11:00
chat with team + Piazza	0:26:00
check specs compliance + check how teammates are doing + Piazza	0:49:00
Creado workspace en togggl con mismo nombre de organización	0:01:59
Creating function crud in workers gui	1:56:30
D03.1: read specs in details and create corresponding issues	0:21:00
D04: read specs + ask questions	0:18:00
D04: read specs + ask questions about these	2:17:00
Debe desarrollarse un conjunto de tests en POSTMAN sobre la API desplegada en heroku que pruebe todas las funcionalidades de la API	2:55:27
Debe tener desplegado en heroku una API REST funcional ofreciendo su fuentes de datos	2:06:34
Desarrollar una página de visualización en modo tabla	1:35:14
desarrollo de un proxy	0:56:36

desarrollo de un proxy	1:20:06
Discuss about communication problems	0:32:00
discuss about Cristina work on date	0:21:00
Feedback on some MA's functions + questions about requirements	0:45:00
fixing the gui of workers data	4:50:58
Font changed #36	0:30:00
full check of spec compliance	0:31:00
gestion de problemas con la machina virtual	0:50:00
governify: testing and bug repports	0:00:48
grabacion L11 + piazza	0:29:00
Improve CityBikes visualization	0:49:00
Investigate and fix some bugs in tools.js with MA	0:57:00
La API debe seguir las guías para fuentes de datos ejemplificadas	2:20:29
La pagina de GUI de gestión debe permitir la siguiente funcionalidades	3:15:32
Modificar archivo index para redirigir todas las API's. #57	4:11:00
modify code of date ready to be launched #34	0:40:00
modify on Heroku some data from my source #24	1:40:00
Olders: formatear la tabla de datos. #77	3:52:00
piazza	3:05:00
Piazza	0:25:00
piazza + coordinate with group for API choice	0:25:00
piazza + final check with test suite	0:35:00
pizza: question about issue management	0:19:00
plan strategy for group part	0:22:00
population GUI: various fixes + check spec compliance	2:03:00
Preparing statistics of Toogle	2:12:00
Questions about PUT	0:07:00
read backlog and plan next tasks	0:16:00
read backlog & extract requierements + run Miguel A. tests	0:22:00
Read backlog + organize work with team	0:21:00
read specs, create issues and planning	0:31:00
Redo lab6 with help of video	0:45:00
refactor CitiBike visualization	0:41:00
Refactor names in HTML + debate about naming conventions	1:44:00
review and discuss about Cristina work on date	0:47:00
Review Miguel Angel work	0:10:00
review REST client with Miguel A.	0:15:00
roundatable: questions + find pricing examples	0:58:00
Se debe tener un enlace dentro la pagina de "/about" a la pagina de visualización de su fuente de datos ( <a href="http://sos-2016-XX.herokuapp.com/data/FFFFFF">http://sos-2016-XX.herokuapp.com/data/FFFFFF</a> )	1:13:04

Tener desplegado en heroku la API en la ruta: "/api/sandbox/XXXXX" siendo XXXXX el recurso ofrecido. #50	1:48:44
Tener desplegado en heroku un enlace dentro la pagina de "/about" a la api de su fuente de datos <a href="http://sos-2016-XX.herokuapp.com/api/v1/FFFFFF">http://sos-2016-XX.herokuapp.com/api/v1/FFFFFF</a> )	0:08:35
Tener guardados en Postman un histórico de peticiones al recurso creado #53	0:40:00
Tener implementado (y desplegado en heroku) una api regulada por SLAs con governify	1:30:38
Test para Miguel A. + Piazza	0:34:00
test REST client + ask confirmation about specs on piazza	0:45:00
trying to optimize code of date. Failed #35	0:52:00
Understand how is working the node js and lunch index.js	2:29:00
Utilizar algun widget (o biblioteca de widgets) para mejorar la apariencia de la tabla de datos que aparece en la dirección: <a href="http://sos-2016-02.herokuapp.com/data/workers">http://sos-2016-02.herokuapp.com/data/workers</a>	1:20:00
Utilizar algun widget (o biblioteca de widgets) para mejorar la apariencia del cliente REST de cada fuente de datos ( <a href="http://sos-2016-XX.herokuapp.com/data/FFFFFF/RESTClient">http://sos-2016-XX.herokuapp.com/data/FFFFFF/RESTClient</a> ), El widget (o biblioteca utilizado) debe ser distinto al de los compañeros de grupo.	1:30:00
Work on Governify	2:00:00