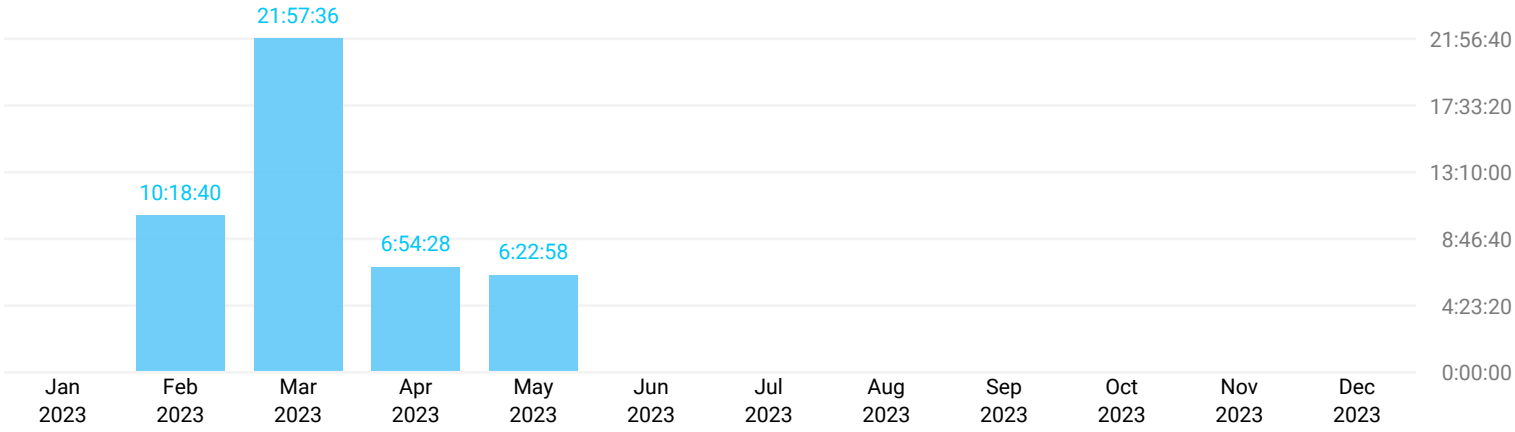
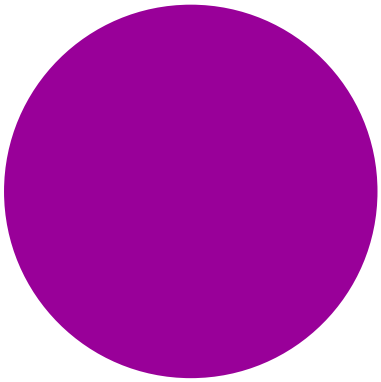
Summary Report

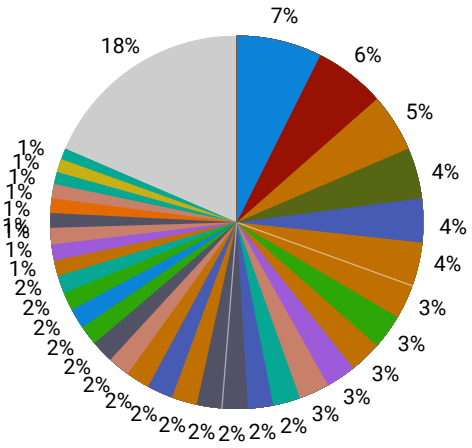
01/01/2023 – 12/31/2023

TOTAL HOURS: 45:33:42





PROJECT	DURATION
SOS2223-24	45:33:42



TIME ENTRY	DURATION
#59 Integrar o usar al menos 4 APIs distintas y que sean distintas a las que han integrado/usado los compañeros del grupo	3:21:20
#35 Búsqueda por todos los campos	2:48:00
#4 Programar "index-YYY.js"	2:19:22
#48 Front-end Svelte que permita todas las funcionalidades básicas de gestión de la información realizando peticiones a la API REST: Una forma de crear recursos. Una forma de listar todos los recursos. Una forma de borrar todos los recursos. Una forma de borrar un recurso concreto. Una forma de editar recursos.	2:00:16
#50 Frontend Svelte	1:42:00
#58 Alguna de las integraciones o usos se debe hacer a través de un proxy propio.	1:42:00
#10 Replicar el algoritmo del archivo llamado “index-YYY.js” dentro del archivo “index.js”	1:23:00
#3 Clonar el repositorio común de la asignatura	1:21:43
#19 Asegurarse que el nombre del recurso (FFFFF) debe cumplir las siguientes restricciones:	1:15:34
#26 Buenas Practicas	1:11:00

#21 Debe desarrollarse una colección de llamadas en POSTMAN sobre la API desplegada en GCloud que pruebe todas las funcionalidades de la API (tal como se vió en el L06)	1:10:55
#8 Archivo index.YYY.js	1:04:42
#12 Verificar que la pestaña individual de la ficha de trabajo cumple los requisitos	1:01:40
#5 Creación index-YYY vicsanesp	1:00:47
#46 Se debe tener una vista correctamente configurada en http://sos2223-XX.appspot.com/ donde debe aparecer como mínimo: Un enlace a cada una de los front-end desarrollados por cada miembro del grupo. Un enlace a cada una de las URL base de las APIs desarrolladas. Un enlace a cada una de las páginas de documentación de Postman de las APIs desarrolladas. Un enlace al repositorio de GitHub del equipo. El nombre de los componentes del equipo y el nombre de su fuente de datos (que debe coincidir con el nombre de recurso de su API).	1:00:06
#47 Un back-end basado en API REST que tenga todas las funcionalidades que se solicitaron en el D01. Si se realiza alguna modificación sobre la API se debe realizar una nueva versión de forma que la API deberá estar disponible en: /api/v2/FFFFF. Se debe tener un nuevo portal público de documentación de la API v2 (generado por Postman) sobre el entorno de GCloud, manteniendo un enlace a ambos portales de documentación en el README.md del proyecto.	1:00:06
#2 Issue vicsanesp 1	0:59:38
#51 Mensaje informativo a usuarios	0:55:00
#27 Deplegar API	0:53:00
#11 Registrar información en el repositorio común de la asignatura	0:51:53
#33 Implementar paginación en la API	0:49:04
#43 Extender coleccion Postman	0:47:00
#37 Los identificadores de recursos que están compuestos por varias propiedades deben estar correctamente utilizados	0:43:29
#57 Uso de alguna biblioteca especificada en la lista awesome-charting o similar	0:41:28
#2 Configuración postman vicsanesp	0:38:36
#56 Uso de algún widget de la biblioteca Highcharts	0:38:10
#31 Implementar la API	0:37:53
#7 Configuracion de Postman	0:35:41
#17 Replicar el algoritmo del archivo llamado “index-YYY.js"	0:34:00
#42 Aádir un script de lanzamiento llamado test-FFFF en el archivo package.json	0:31:00
#39 Se debe devolver en cada caso un Objeto o un Array en función del tipo de operación que se realice	0:30:57
#9 Mantener al dia y correcion de errores anteriores	0:30:00
#1 Configurar POSTMAN	0:28:53
Other time entries	8:25:29

PROJECT - TIME ENTRY	DURATION	PERCENTAGE
SOS2223-24	45:33:42	100.0%
#1 Configurar POSTMAN	0:28:53	1.06%

PROJECT - TIME ENTRY	DURATION	PERCENTAGE
#10 Replicar el algoritmo del archivo llamado “index-YYY.js” dentro del archivo “index.js”	1:23:00	3.04%
#11 Registrar información en el repositorio común de la asignatura	0:51:53	1.9%
#12 Verificar que la pestaña individual de la ficha de trabajo cumple los requisitos	1:01:40	2.26%
#13 Replicar el algoritmo del archivo llamado “index-YYY.js	0:15:07	0.55%
#14 Realizar tracking de tiempo con Toggl de todas las issues asignadas	0:15:05	0.55%
#15 Haber leído todas las noticias en Piazza previas a 16 horas antes del inicio de la sesión de feedback.	0:15:18	0.56%
#16 Registrar información en el repositorio común de la asignatura	0:15:32	0.57%
#17 Replicar el algoritmo del archivo llamado “index-YYY.js"	0:34:00	1.24%
#18 Tener una ruta dinámica “/cool”	0:18:12	0.67%
#19 Asegurarse que el nombre del recurso (FFFFF) debe cumplir las siguientes restricciones:	1:15:34	2.76%
#2 Configuración postman vicsanesp	0:38:36	1.41%
#2 Issue vicsanesp 1	0:59:38	2.18%
#21 Debe desarrollarse una colección de llamadas en POSTMAN sobre la API desplegada en GCloud que pruebe todas las funcionalidades de la API (tal como se vió en el L06)	1:10:55	2.59%
#22 Debe tener desplegado en Google Cloud una API REST funcional ofreciendo su fuente de datos. La API debe estar desplegada (e integrada con los compañeros de grupo) en la dirección: http://sos2223-XX.appspot.com/api/v1/FFFFF (Siendo XX el numero de grupo relleno con ceros y FFFFF el nombre del recurso).	0:20:20	0.74%
#23 El recurso debe contener una ruta /api/v1/FFFFF/loadInitialData que al hacer un GET cree 10 o más datos en el array de NodeJS si está vacío.	0:20:10	0.74%
#24 La API debe cumplir con las buenas prácticas definidas en los laboratorios: Deben implementarse todos los métodos de la tabla azul (vistos en el L05) Deben usarse todos los códigos de estado del cuadro verde (vistos en el L05) No se debe devolver HTML en ningún caso	0:20:10	0.74%

PROJECT - TIME ENTRY	DURATION	PERCENTAGE
#25 LoadInitialData	0:27:00	0.99%
#26 Buenas Practicas	1:11:00	2.6%
#27 Deplegar API	0:53:00	1.94%
#28 La API debe implementar una persistencia basada en una base de datos NeDB tal como se vio en el L07.	0:20:09	0.74%
#29 La API debe implementar búsquedas por todos los campos del recurso (siguiendo las buenas prácticas recomendadas en el L06)	0:20:05	0.73%
#3 Clonar el repositorio común de la asignatura	1:21:43	2.99%
#3 Instalar y configurar extensión de Chrome de Toggl	0:11:02	0.4%
#30 El proyecto debe contener una ruta /api/v1/FFFFF/docs que redirija al portal de documentación generado en POSTMAN (tal como se vió en el L07)a partir de la colección de llamadas del punto 6. Para redirigir se recomienda utilizar el método “redirect” (ejemplo).	0:20:06	0.74%
#31 Implementar la API	0:37:53	1.39%
#32 La API debe implementar búsquedas por todos los campos del recurso	0:25:40	0.94%
#33 Implementar paginación en la API	0:49:04	1.79%
#34 El proyecto debe contener una ruta /api/v1/FFFFF/docs	0:27:00	0.99%
#35 Búsqueda por todos los campos	2:48:00	6.15%
#36 Implementar NeDB	0:18:00	0.66%
#37 Los identificadores de recursos que están compuestos por varias propiedades deben estar correctamente utilizados	0:43:29	1.59%
#38 Extender la colección de llamadas de POSTMAN (para que funcionen, al menos, sobre la api desplegada en GCloud) de forma que cada llamada debe tener siempre alguna comprobación y, en los casos que sea posible, que vaya más allá de comprobar el estado (p.e. comprobar que el dato devuelto tenga algún campo o que el array devuelto tenga un tamaño determinado)	0:20:06	0.74%

PROJECT - TIME ENTRY	DURATION	PERCENTAGE
#39 Se debe devolver en cada caso un Objeto o un Array en función del tipo de operación que se realice	0:30:57	1.13%
#4 Programar "index-YYY.js"	2:19:22	5.1%
#40 Se debe añadir un script de lanzamiento llamado test-FFFF en el archivo package.json que lance (tal como se vió en el L08), con newman, la colección de test de la API /api/v1/FFFFF desplegada en local de manera que se pueda ejecutar con el comando "npm run test-FFFFF".	0:26:04	0.95%
#41 Tener configurado el sistema de integración continua tal como se vio en el L08 de forma que al hacer push al repositorio se lancen las pruebas automáticamente sobre todas las apis con el comando "npm test" y que el estado antes de la clase de feedback sea de correcto y aparezca el símbolo de check válido verde (✅) al lado del identificador del último commit (que no sea "Merge pull Request").	0:20:05	0.73%
#42 Aádir un script de lanzamiento llamado test-FFFF en el archivo package.json	0:31:00	1.13%
#43 Extender coleccion Postman	0:47:00	1.72%
#44 Añadir run test-FFFF	0:20:00	0.73%
#45 Identificadores de recursos	0:25:00	0.91%
#46 Se debe tener una vista correctamente configurada en http://sos2223-XX.appspot.com/ donde debe aparecer como mínimo: Un enlace a cada una de los front-end desarrollados por cada miembro del grupo. Un enlace a cada una de las URL base de las APIs desarrolladas. Un enlace a cada una de las páginas de documentación de Postman de las APIs desarrolladas. Un enlace al repositorio de GitHub del equipo. El nombre de los componentes del equipo y el nombre de su fuente de datos (que debe coincidir con el nombre de recurso de su API).	1:00:06	2.2%
#47 Un back-end basado en API REST que tenga todas las funcionalidades que se solicitaron en el D01. Si se realiza alguna modificación sobre la API se debe realizar una nueva versión de forma que la API deberá estar disponible en: /api/v2/FFFFF. Se debe tener un nuevo portal público de documentación de la API v2 (generado por Postman) sobre el entorno de GCloud, manteniendo un enlace a ambos portales de documentación en el README.md del proyecto.	1:00:06	2.2%
#48 Front-end Svelte que permita todas las funcionalidades básicas de gestión de la información realizando peticiones a la API REST: Una forma de crear recursos. Una forma de listar todos los recursos. Una forma de borrar todos los recursos. Una forma de borrar un recurso concreto. Una forma de editar recursos.	2:00:16	4.4%
#49 No Json ni codigos de estado	0:17:00	0.62%
#5 Creación index-YYY vicsanesp	1:00:47	2.22%

PROJECT - TIME ENTRY	DURATION	PERCENTAGE
#5 Realizar tracking de tiempo	0:13:07	0.48%
#50 Frontend Svelte	1:42:00	3.73%
#51 Mensaje informativo a usuarios	0:55:00	2.01%
#56 Uso de algún widget de la biblioteca Highcharts	0:38:10	1.4%
#57 Uso de alguna biblioteca especificada en la lista awesome-charting o similar	0:41:28	1.52%
#58 Alguna de las integraciones o usos se debe hacer a través de un proxy propio.	1:42:00	3.73%
#59 Integrar o usar al menos 4 APIs distintas y que sean distintas a las que han integrado/usado los compañeros del grupo	3:21:20	7.36%
#6 Clonar repositorio de la asignatura vicsanesp	0:22:53	0.84%
#6 trackeo de toggl con git	0:15:08	0.55%
#7 Configuracion de Postman	0:35:41	1.31%
#8 Archivo index.YYY.js	1:04:42	2.37%
#9 Mantener al dia y correcion de errores anteriores	0:30:00	1.1%
arreglando toggl	0:17:08	0.63%
Without description	0:00:02	