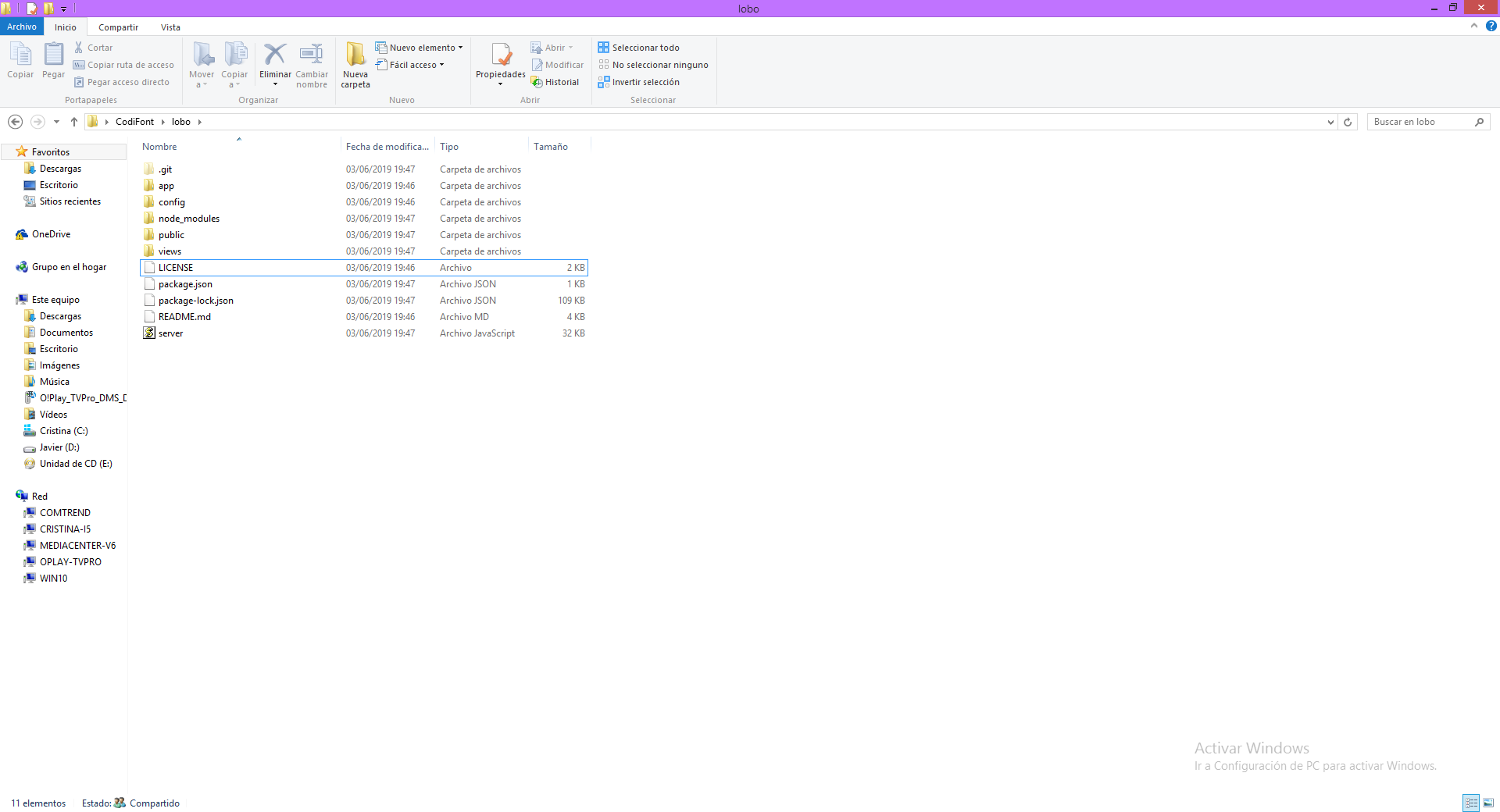
Carpeta principal del programa:

**.git:** Arxius necessaris de Github.

**app:** Conté dues coses. Routes y models.

**config:** Configuració del passport. Aquí conté la config de autentificació, el access al MongoDB i l’arxiu de config del passport.

**node\_modules:** Llibreries i arxius necessaris per el node

**public:** Carpeta que conté els scripts, imatges, etc necessaris per la web.

**views:** Vistes de l’aplicació. Aquí es guarden tots els .ejs (els arxius html web)

**LICENSE:** Licencia del PassportJS. Una llibreria utilitzada pel control d’usuaris

**package.json:**

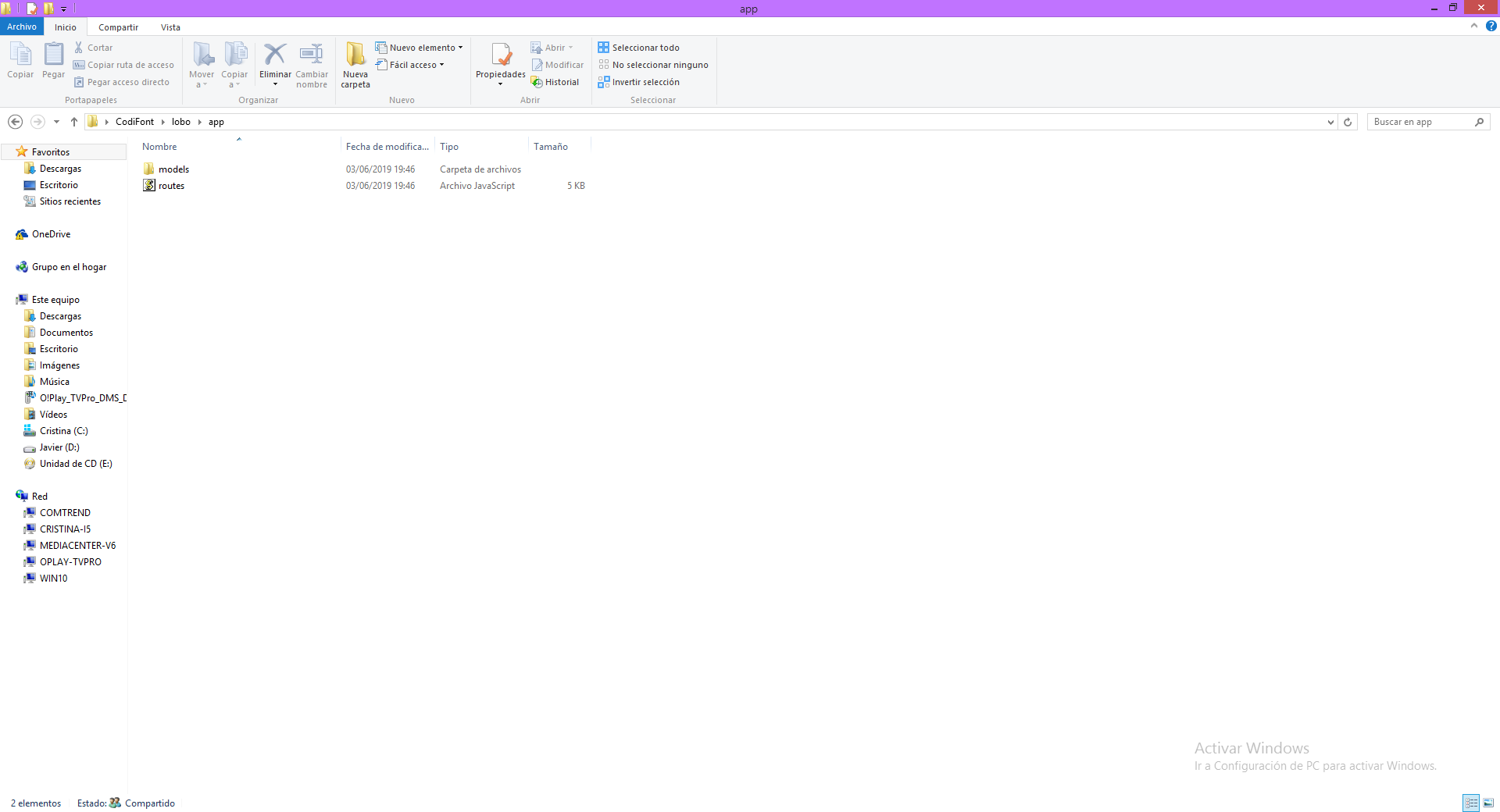
**package-lock.json:**

**README.md:** Readme del projecte amb la informació del joc, com jugar, etc.

**server.js:** Servidor NodeJS que serveix la web i controla el funcionament de la web.

Carpetes desglossades:

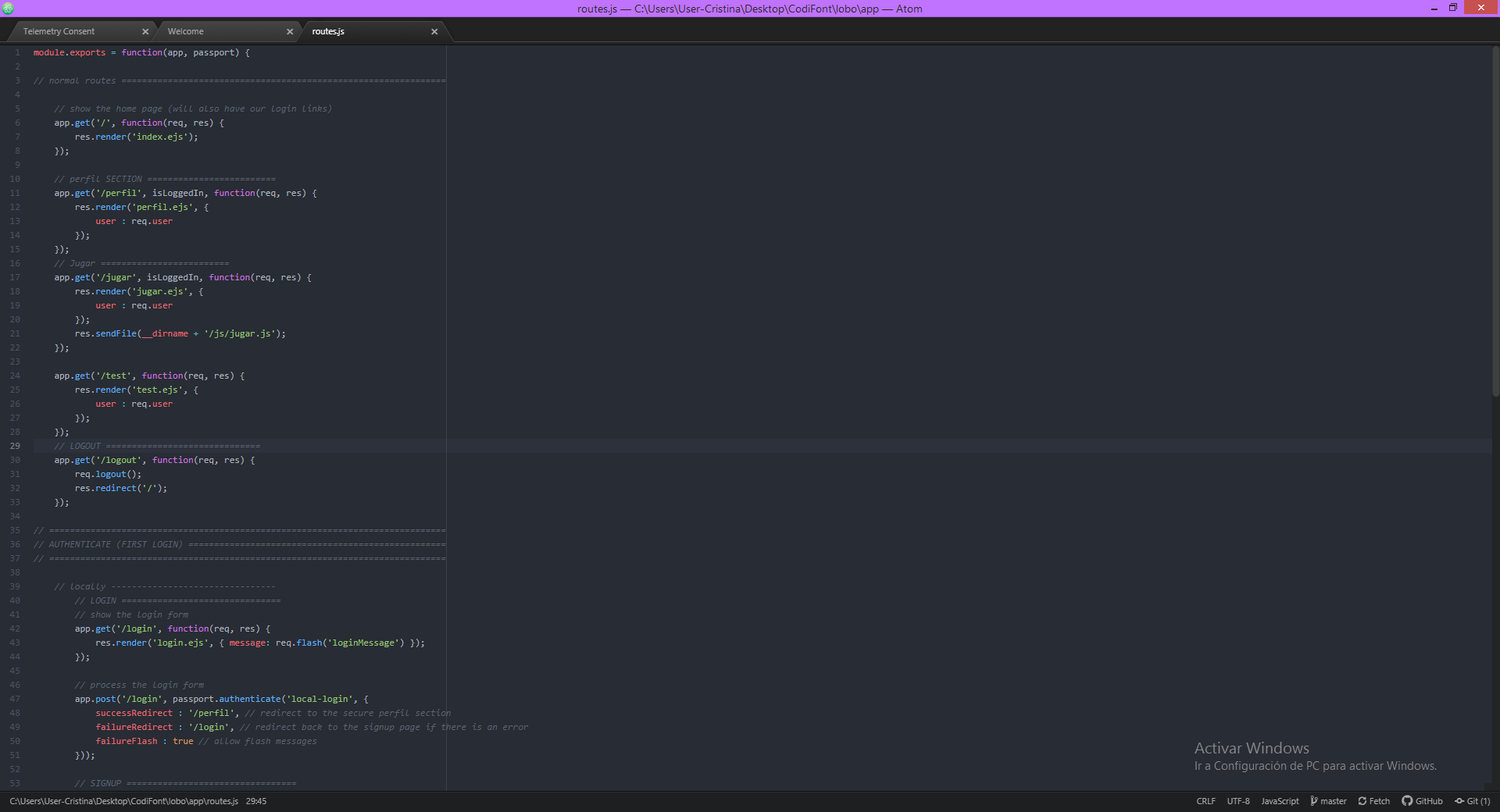
**APP**



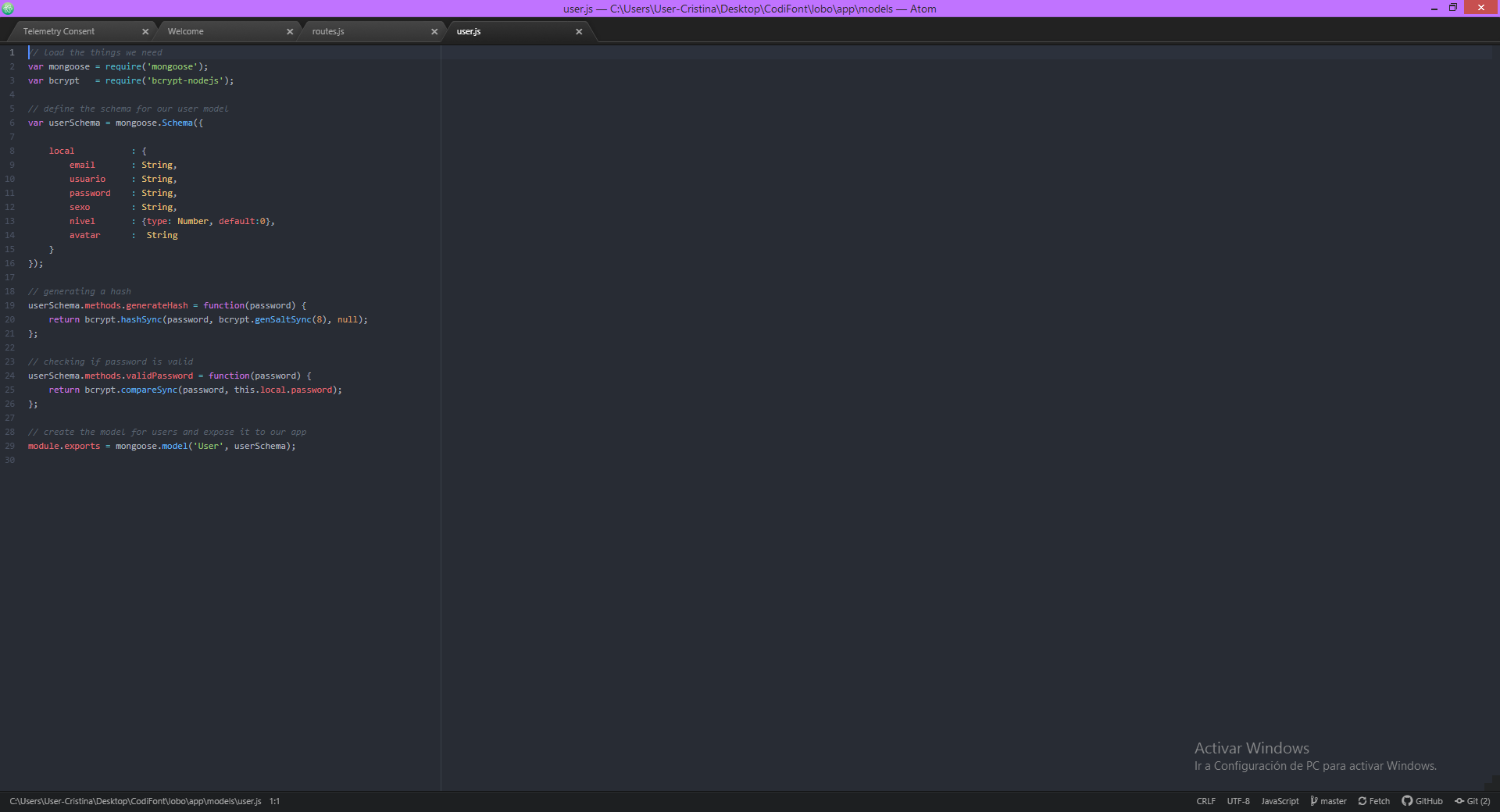
***Routes:*** Aquest és l’arxiu de configuració de les rutes de la nostra web. Utilitzem la llibreria Express de node per fer-ho funcionar.

Aquí definim la url, els paràmetres que passem i quina vista te que carregar.

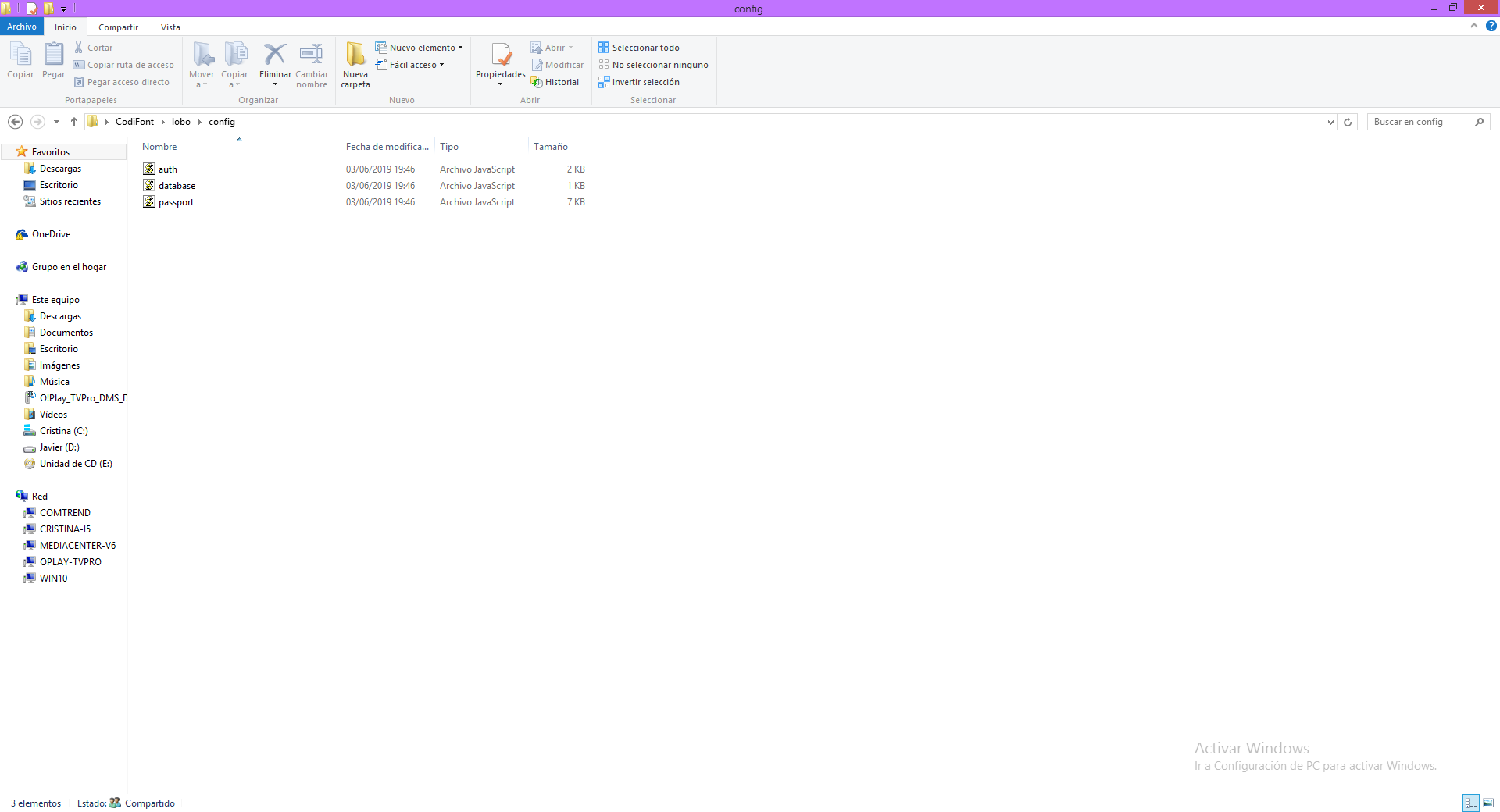
També podem pasar la funció isLoggedIn, encarregada de controlar que el usuari estigui registrat. Si no ho està, no el deixa accedir-hi.



**Models:** Aquesta carpeta conté el fitxer de configuració de la taula usuaris al MongoDB que utilitza el PassportJS. Un cop registrem un usuari, omplirá aquest camps i el guardará a la BD interna.

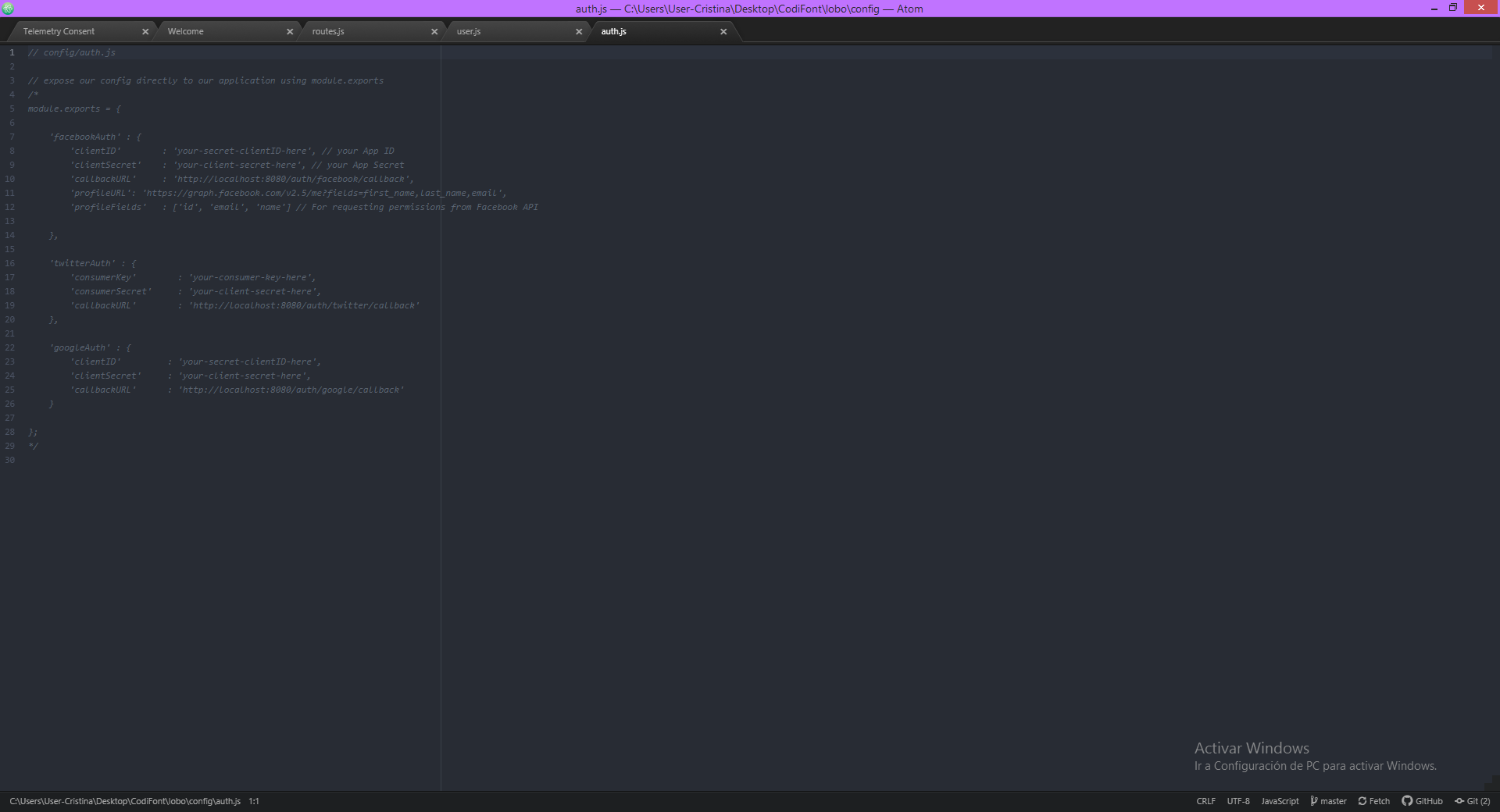


**Config**

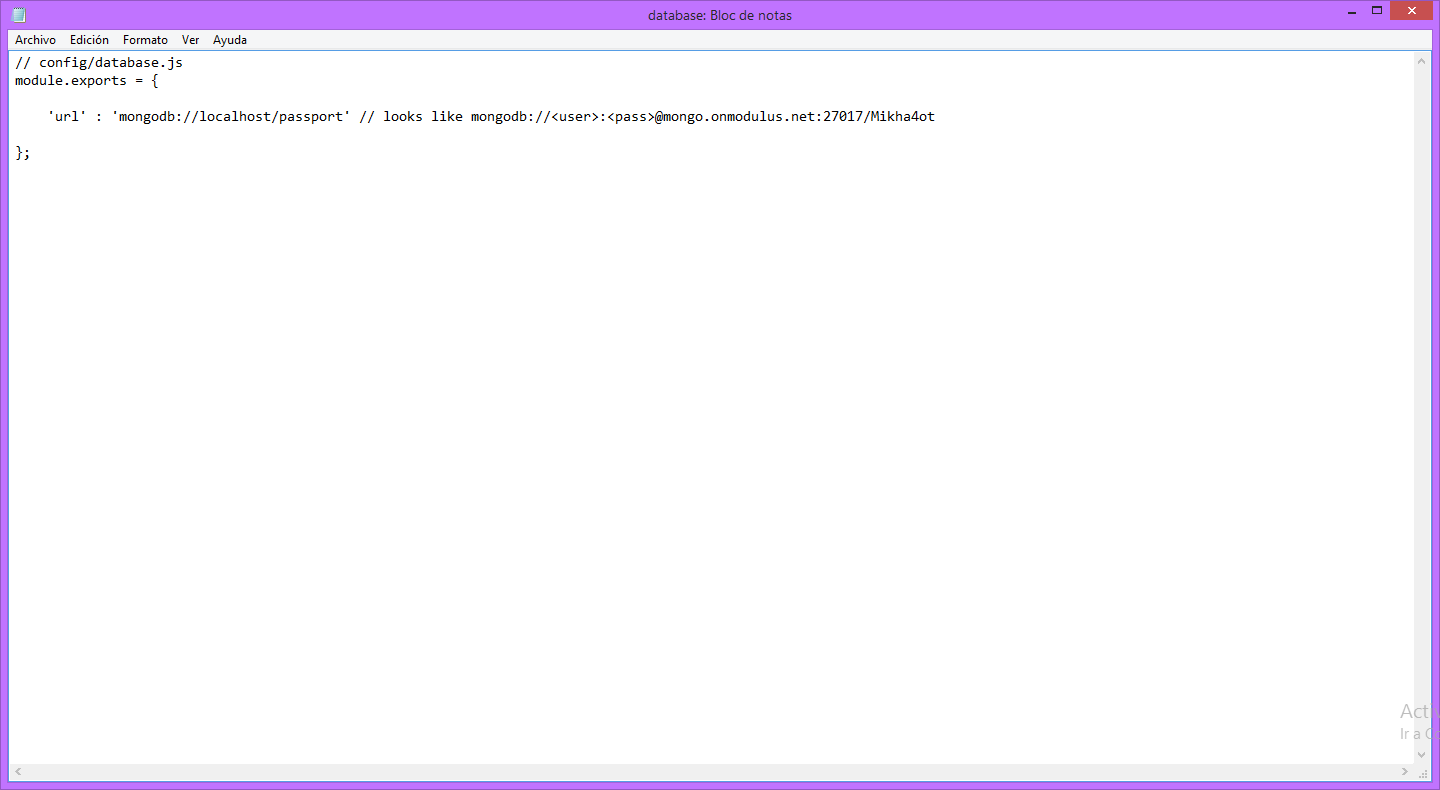


Es la carpeta de configuració del PassportJS, l’access a MongoDB i la configuració de autentificació.

***Auth:***Fitxer de configuració de l'autenticació. Tenim, a part de l’access local, access per FB, Twitter i Google. Actualment està desactivat per no controlar més coses de les necessàries.

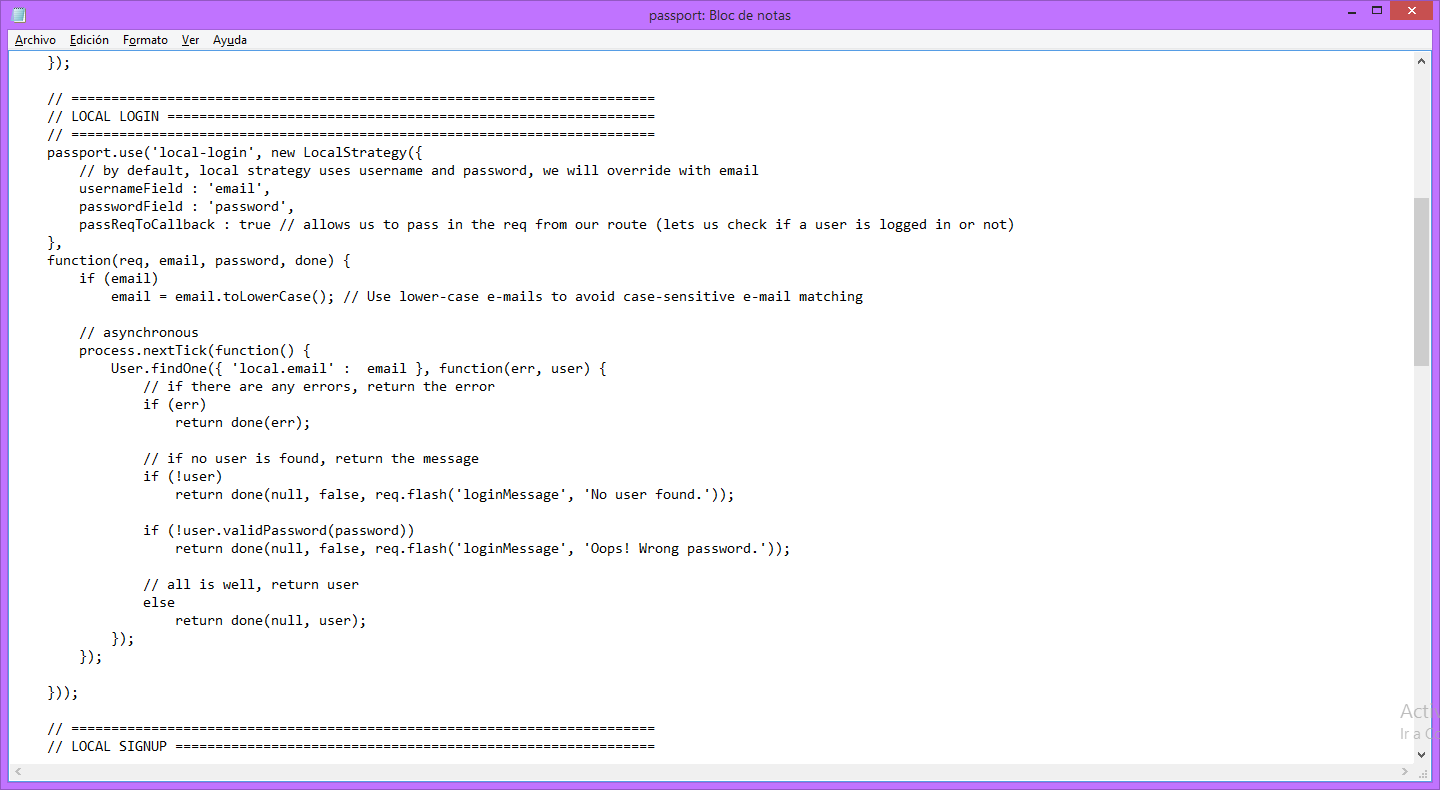


***Database:*** Arxiu de configuració de l'accés al MongoDB amb les credencials.

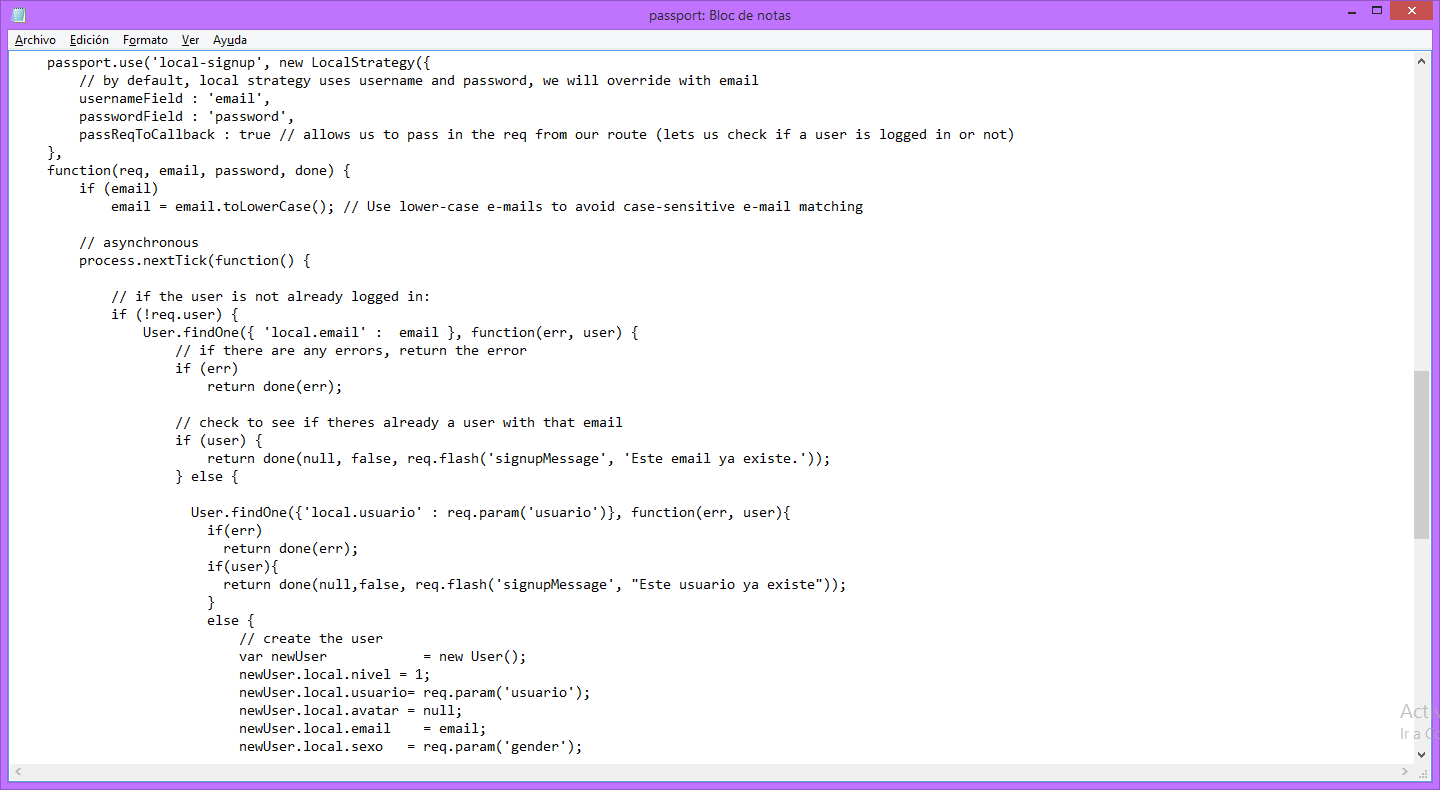


***Passport:***Fitxer que conté i controla el access i registre d’usuaris. Es el fitxer base de la llibreria PassportJS.

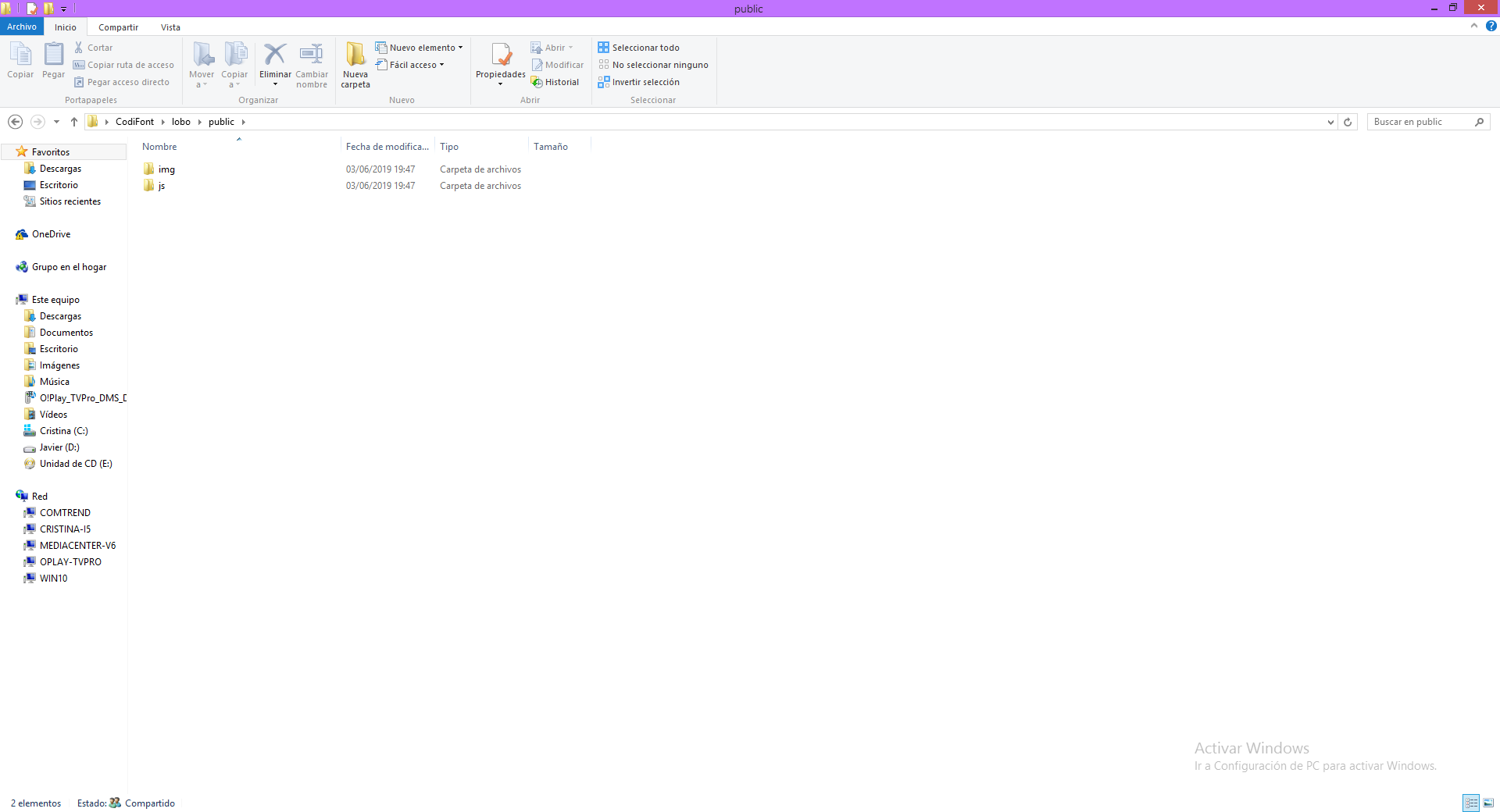
* Login:



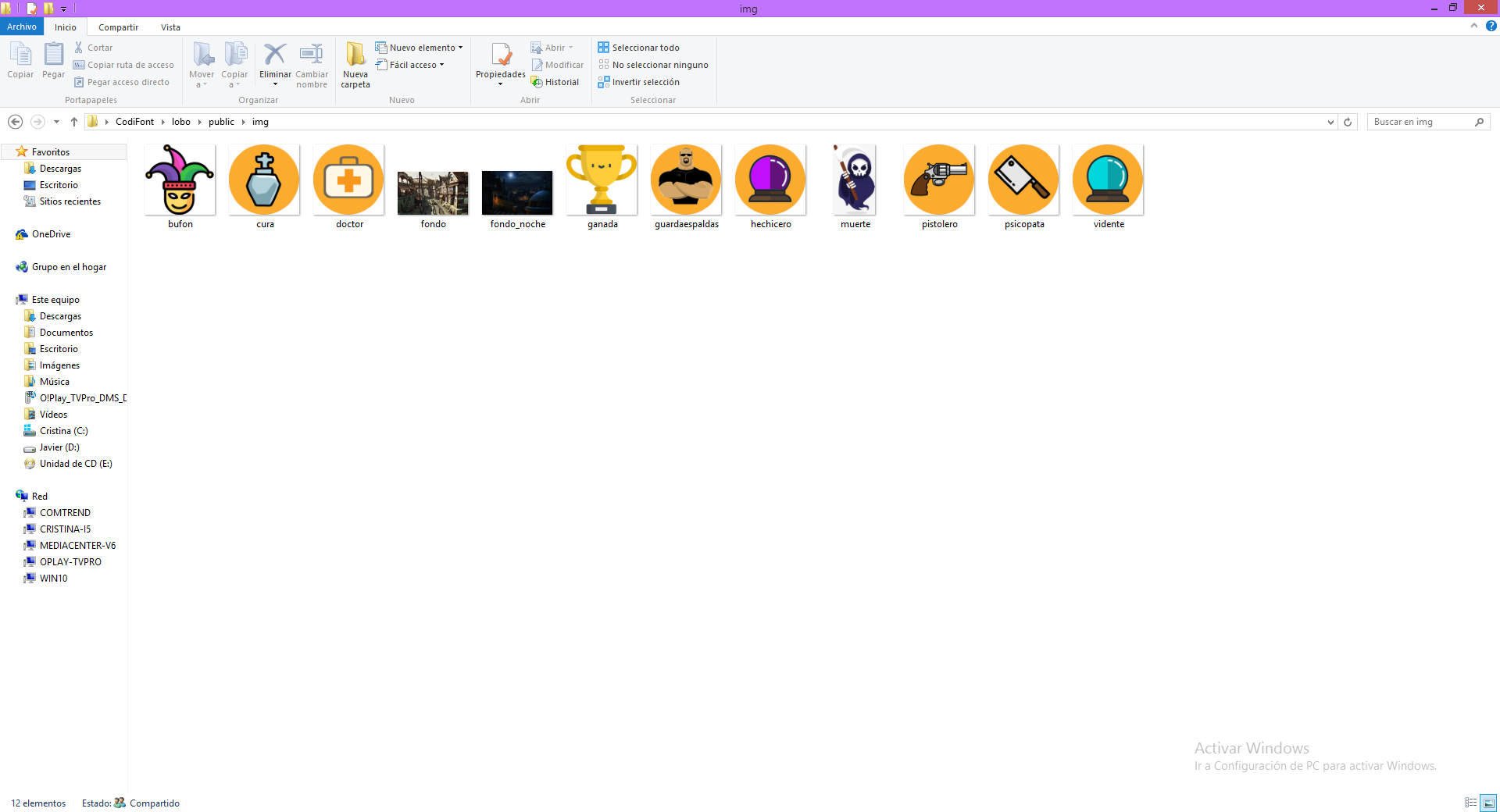
* Registre:

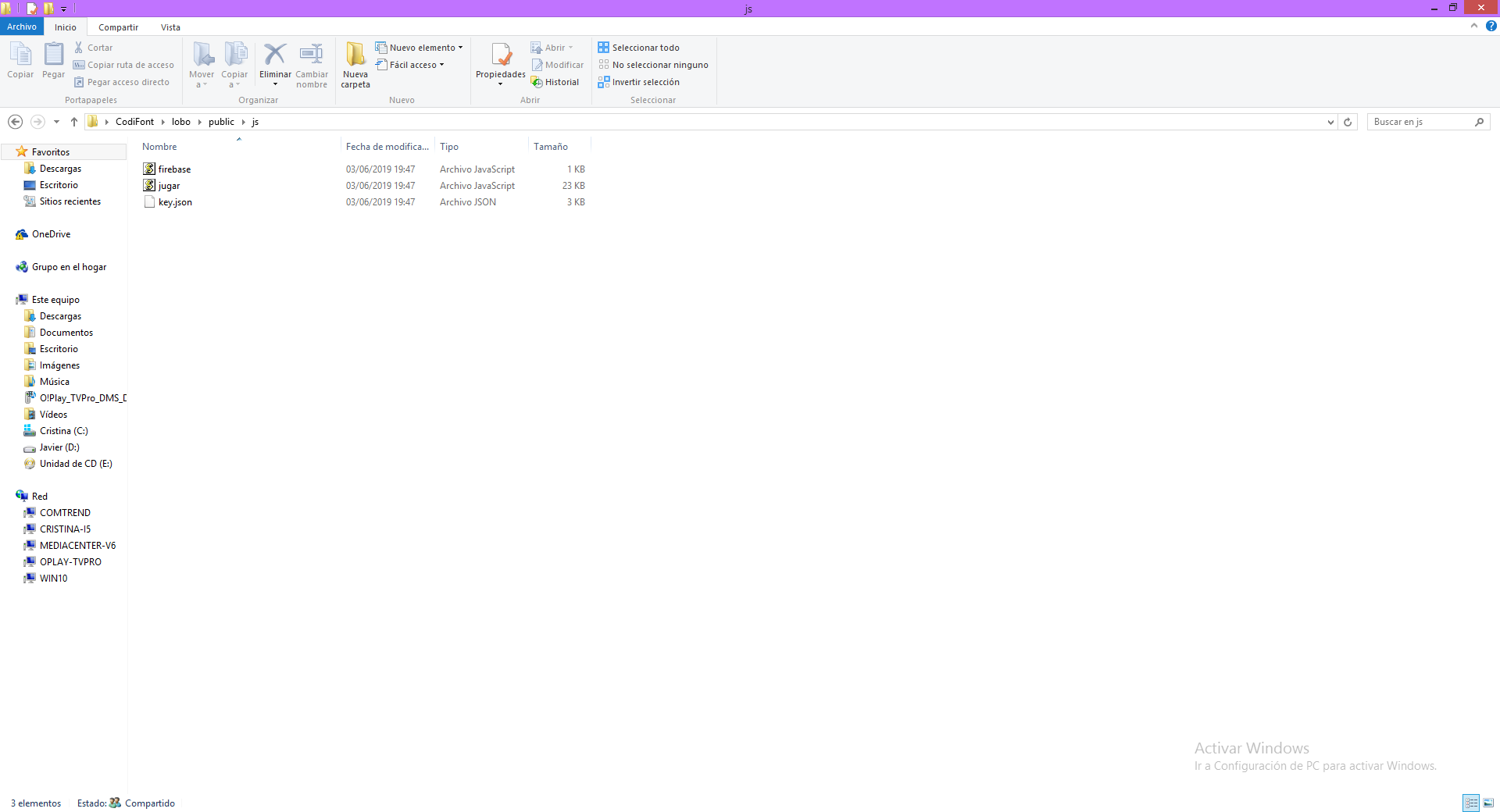


**Public**

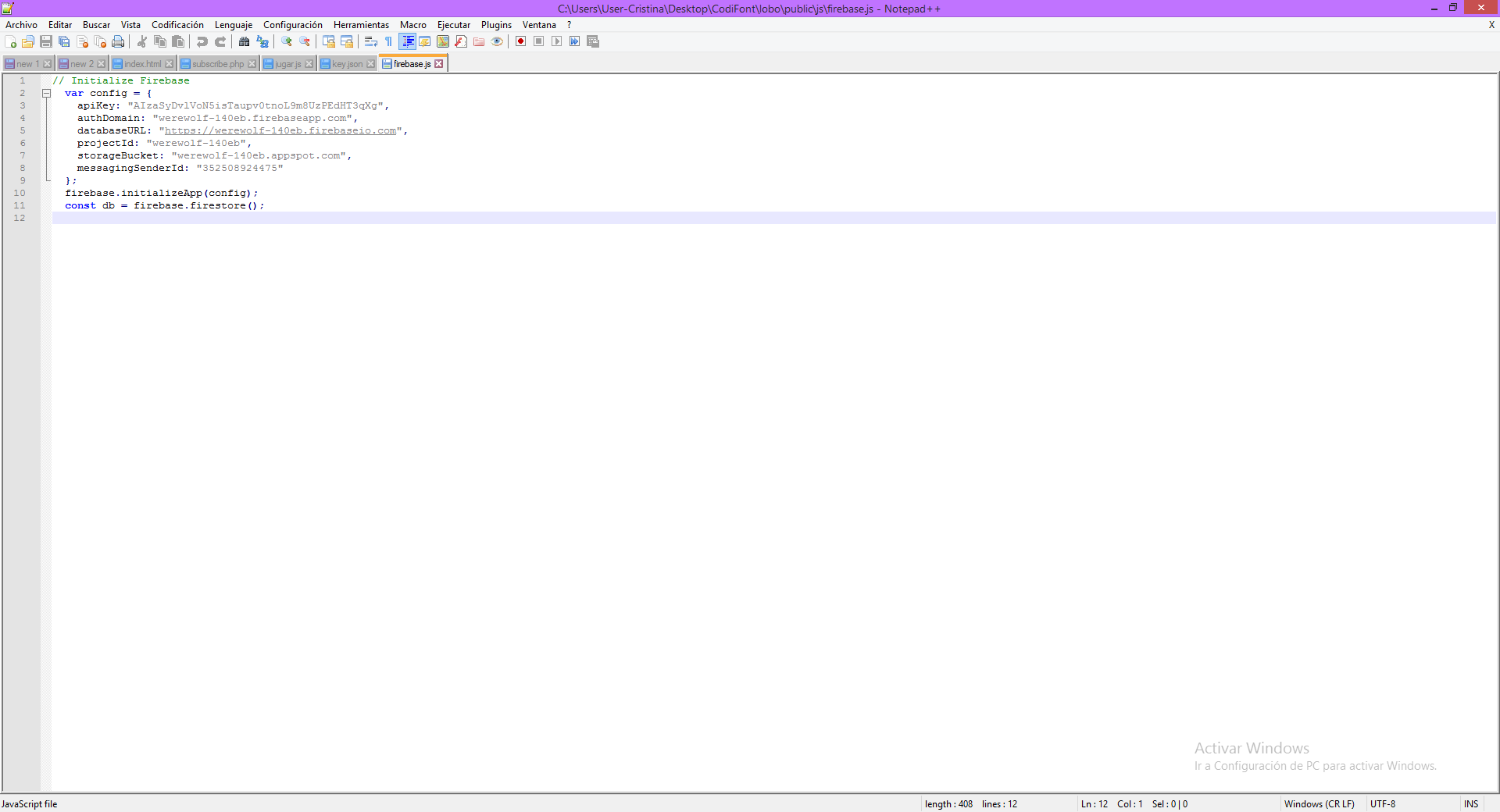


Carpeta amb les imatges i scripts necessaris per la web.

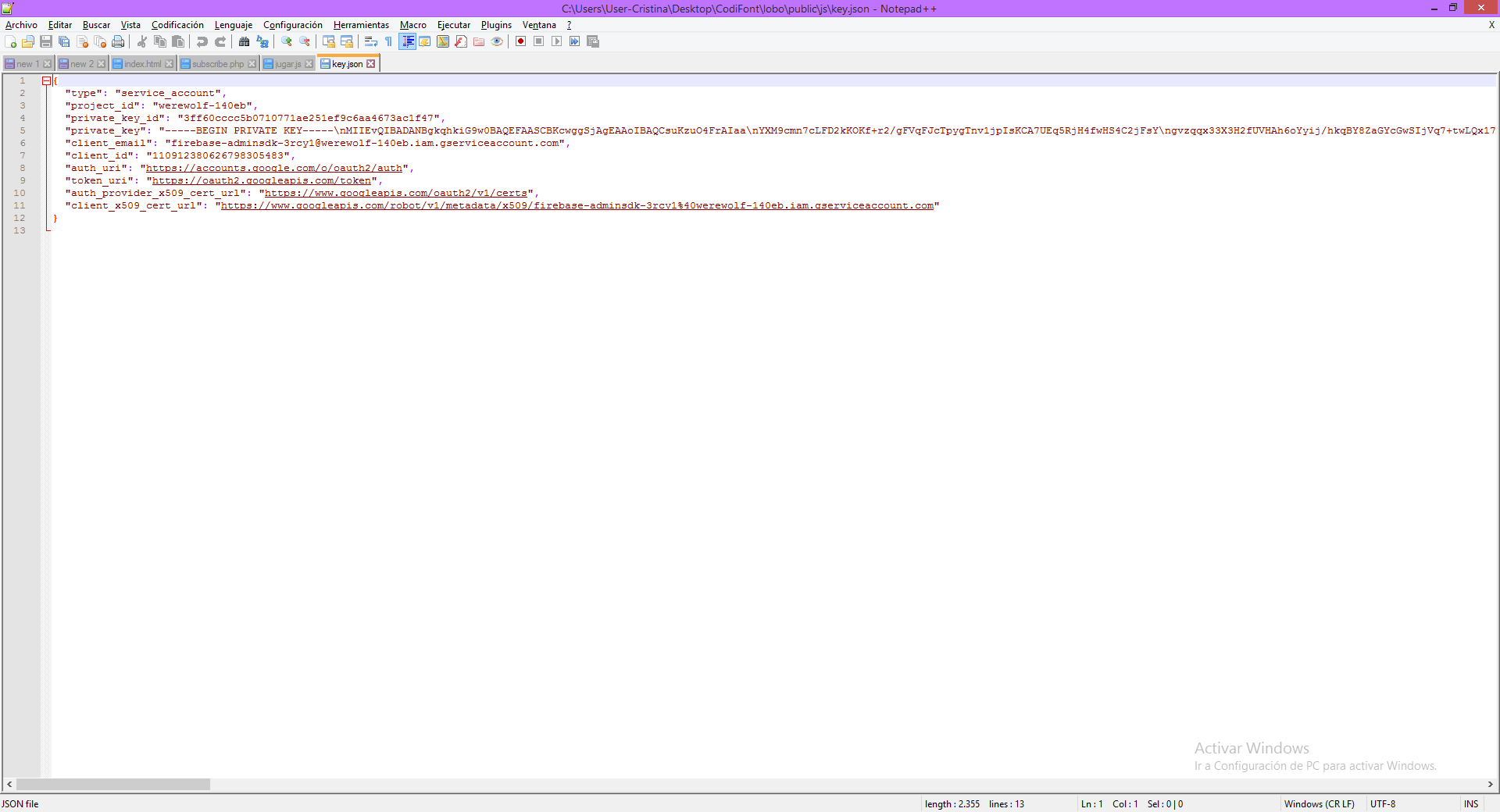
***img:*** Aquesta carpeta conté totes les imatges de la web, imatges d’alertes, de accions de rols, fons de pantalla….****

***js:*** Scripts i accés al Firebase

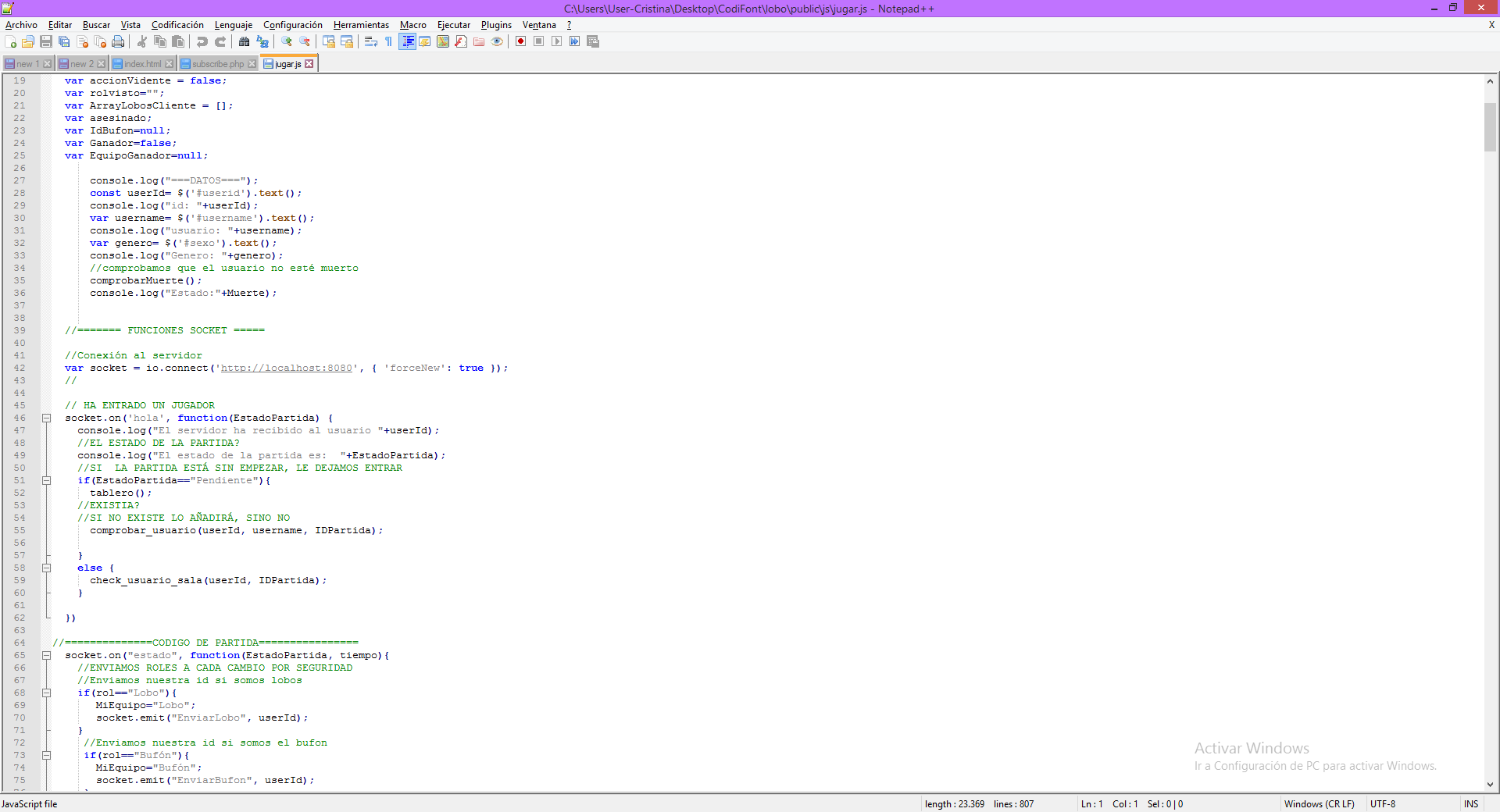
***firebase***: Arxiu amb la configuració d'accés al Firebase i la variable “db” que utilitzem als nostres codis.



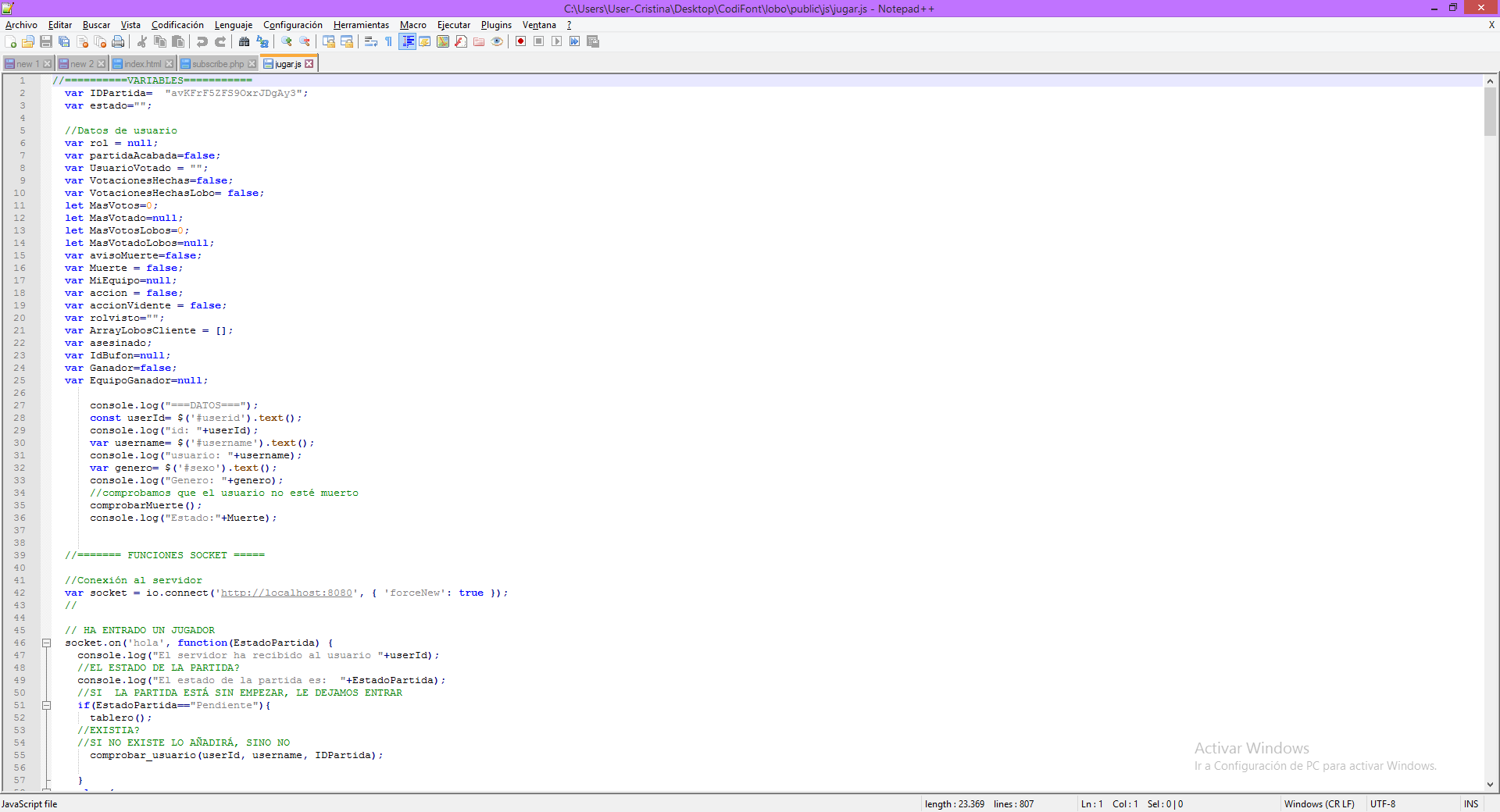
***key.json:*** Key personal del projecte necessesaria per la conexió del NodeJS amb el Firebase.



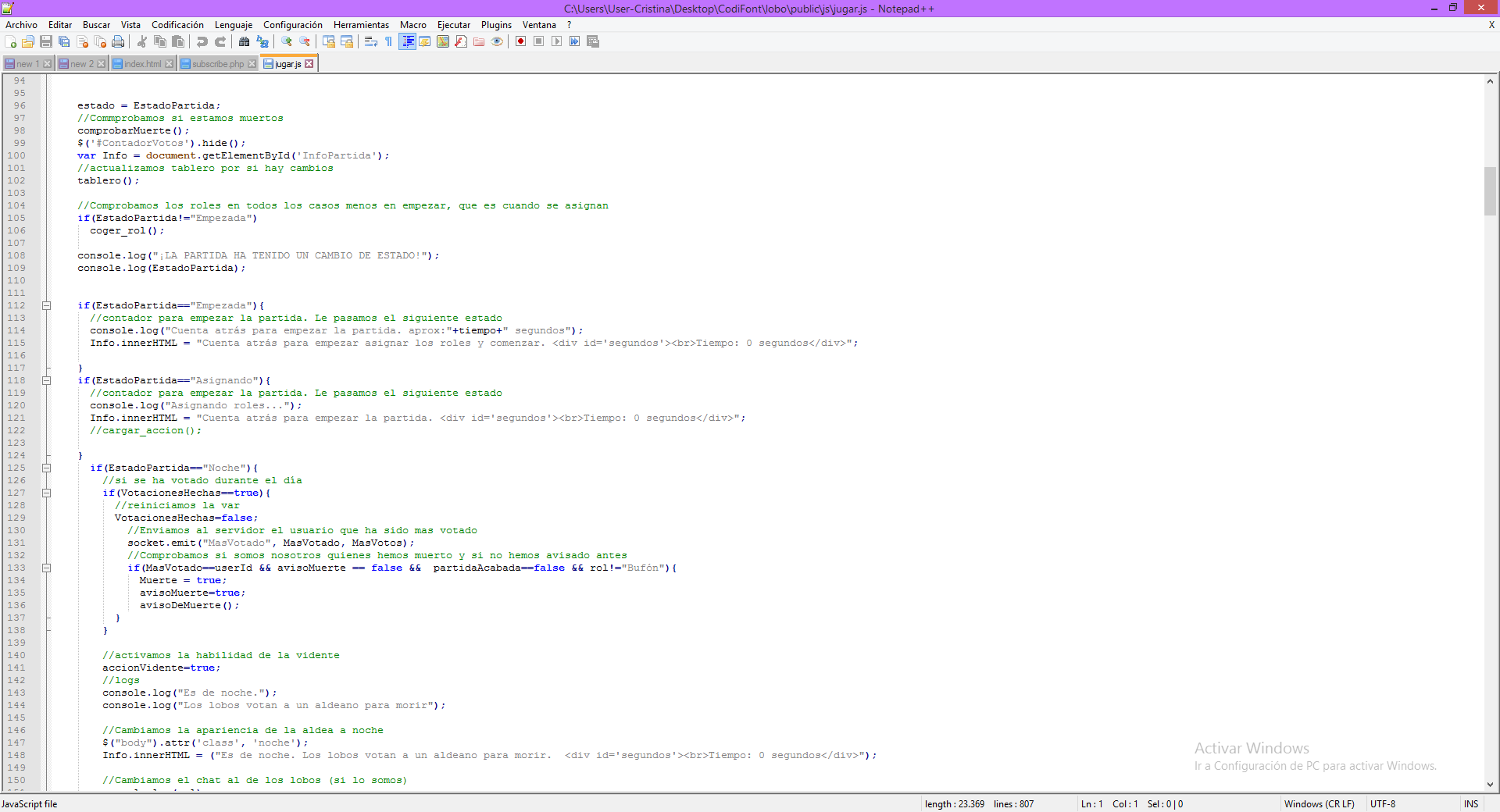
***Jugar.js:*** Es l’arxiu base al nostre joc per a la part de client. Un cop accedim a la pestaña “Jugar” començará a fer la conexió amb Sockets i comprobará l’usuari.



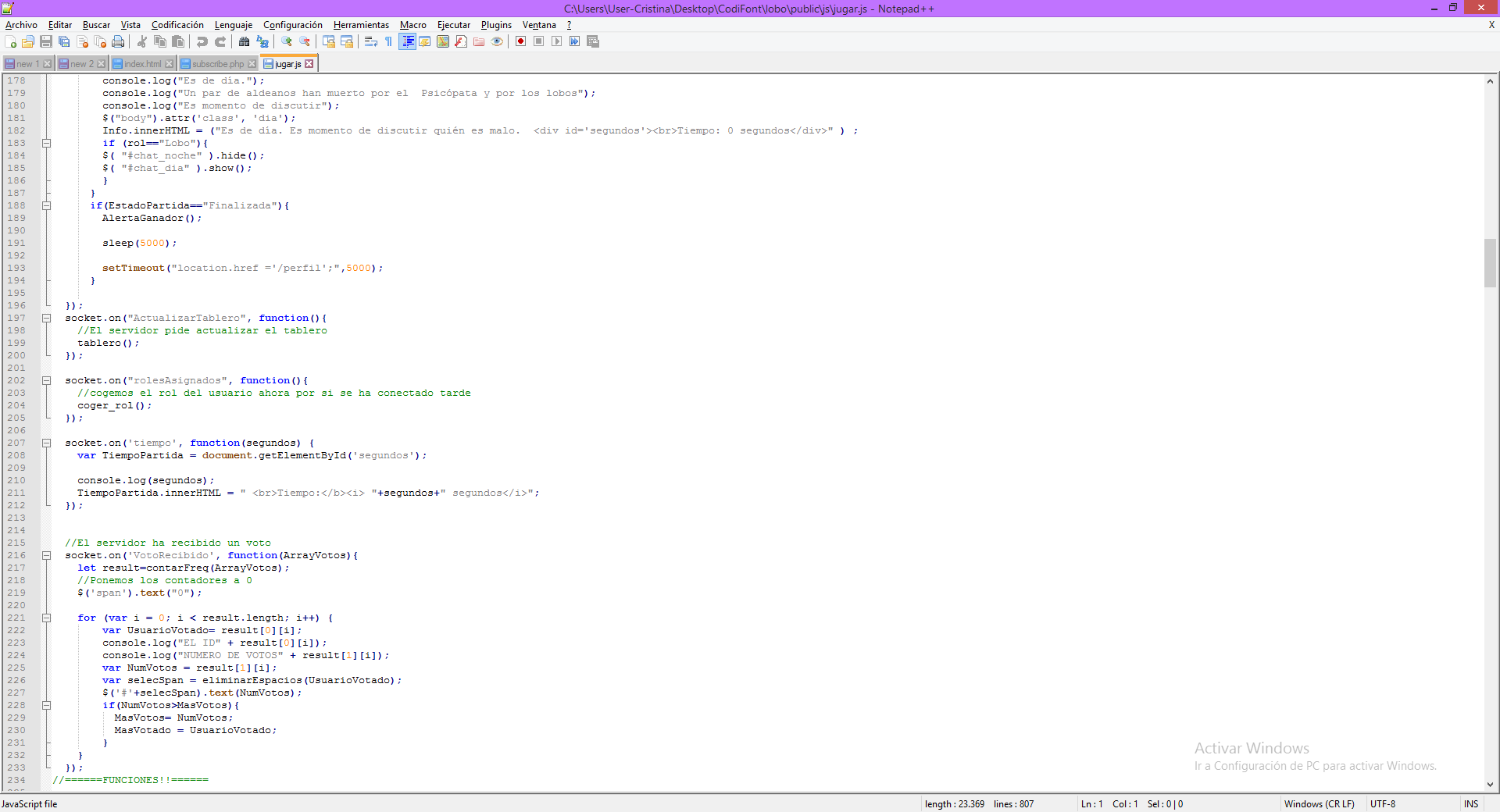
També tenim definides totes les variables necessàries per al funcionament del joc (part client)



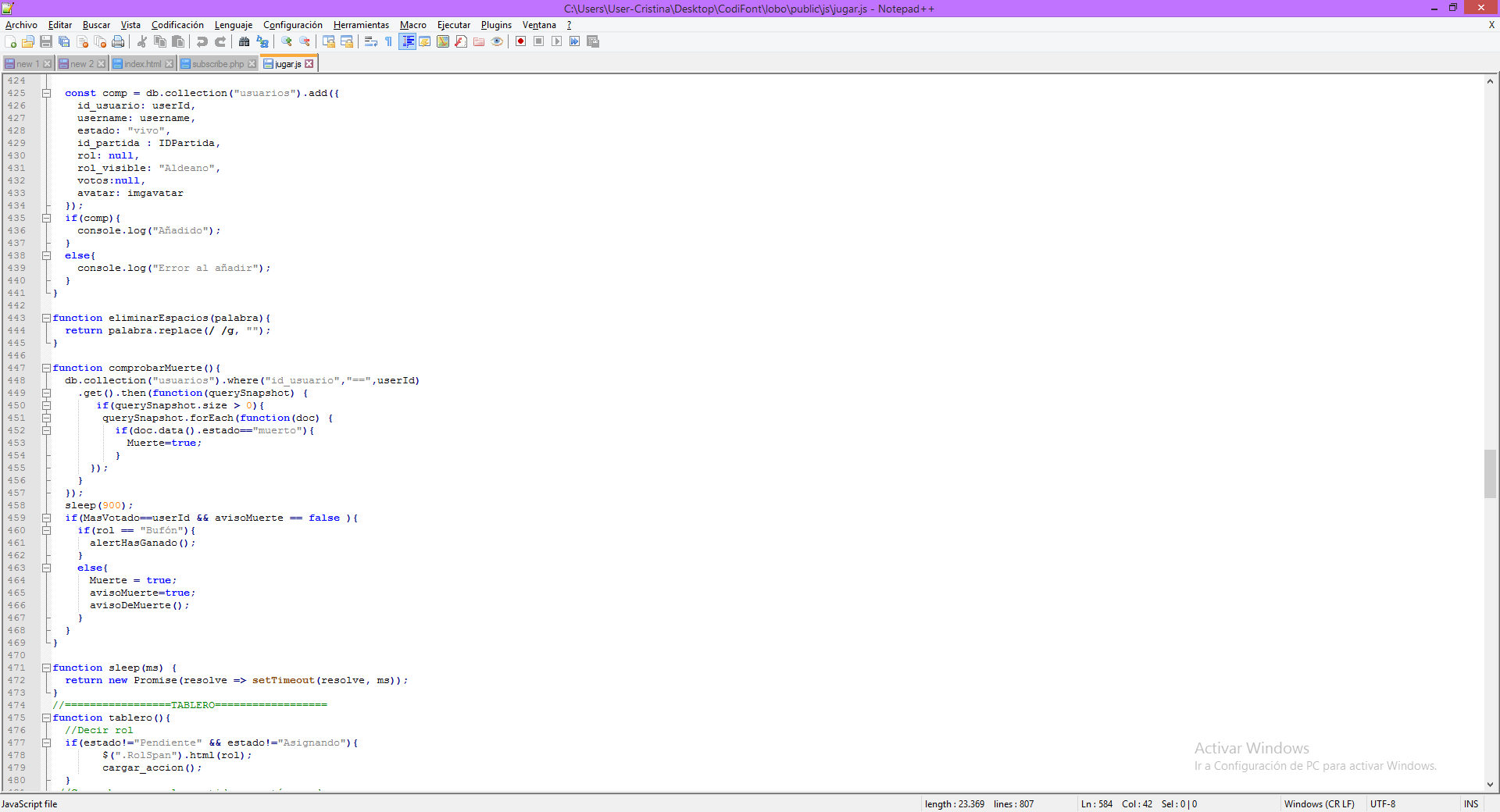
Controla el cicle de nit/día/votació de la part client. Sempre ho comprova amb el servidor per evitar un “desfase” o trampes al actualizar o tancar.



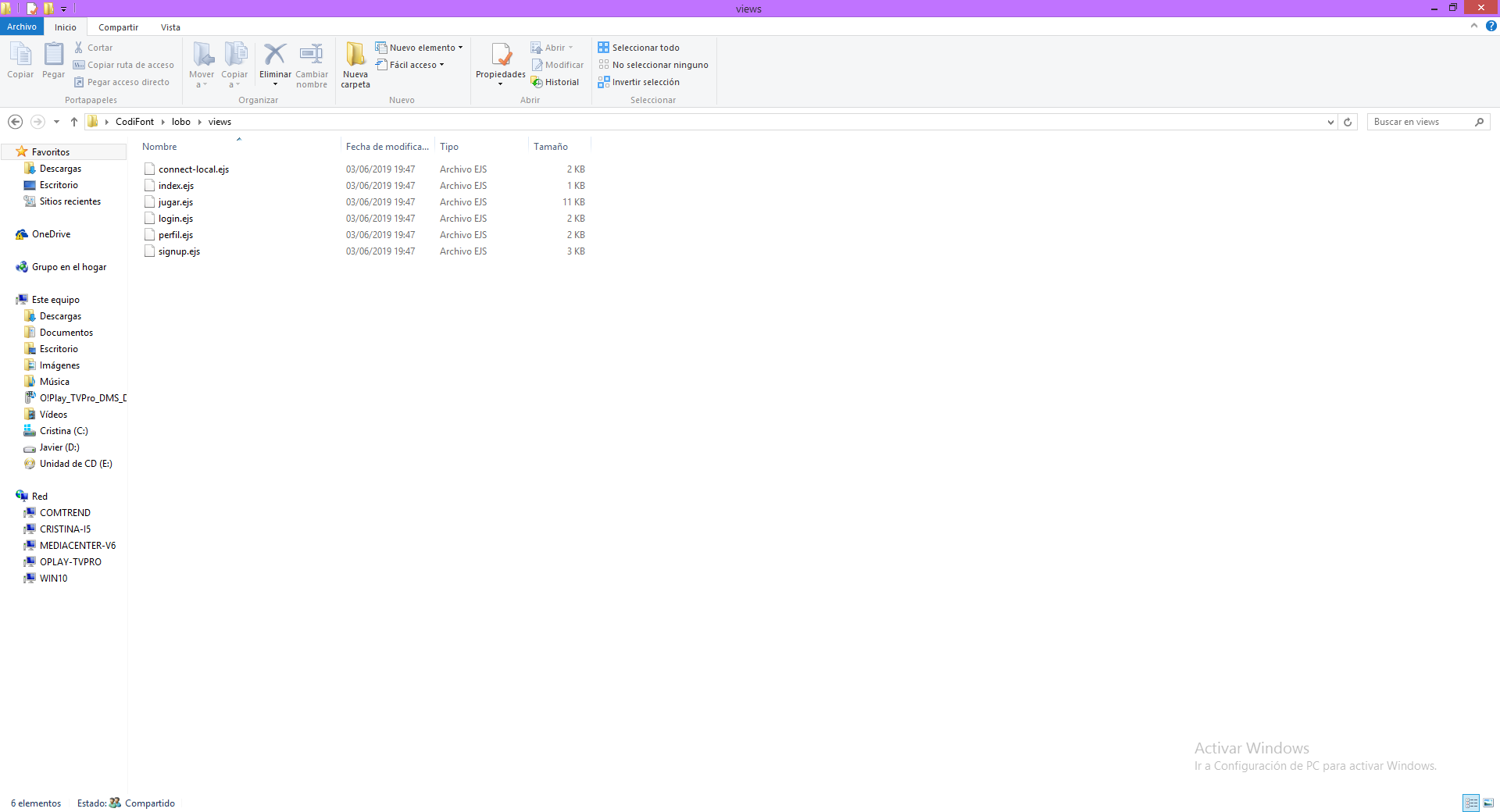
Controla els sockets que envia el servidor al client i executa la funció de cadascun d’ells. Exemples:



També conté les alertes amb SweetAlert, funcions necessesaries per a les accions de rol, per al desenvolupament de la partida i actualitzacions a la BD de Firebase.



***views:*** Directori amb les vistes de la web. Cada una de les vistes es un apartat.



***connect-local.ejs:*** Arxiu al que s’accedeix si intentes mouret sense registre.

***index.ejs:*** Página principal de la web.

***jugar.ejs:*** Página de joc. Conté el taulell de joc, el xat i la part visual de la partida.

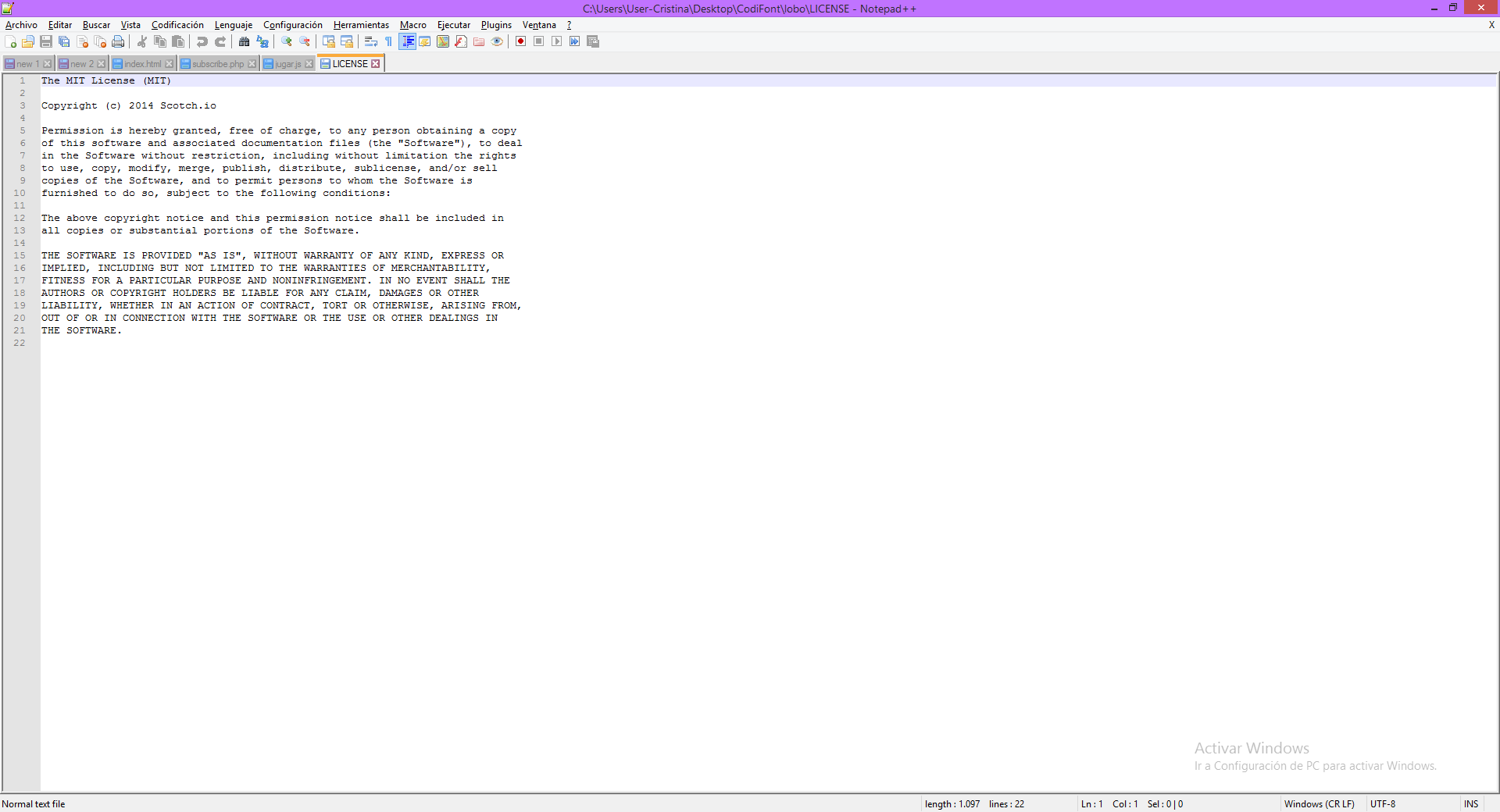
***login.ejs:*** Login web.

***perfil.ejs:*** Perfil provisional de l’usuari on es pot veure la informació

***signup.ejs:*** Registre web.

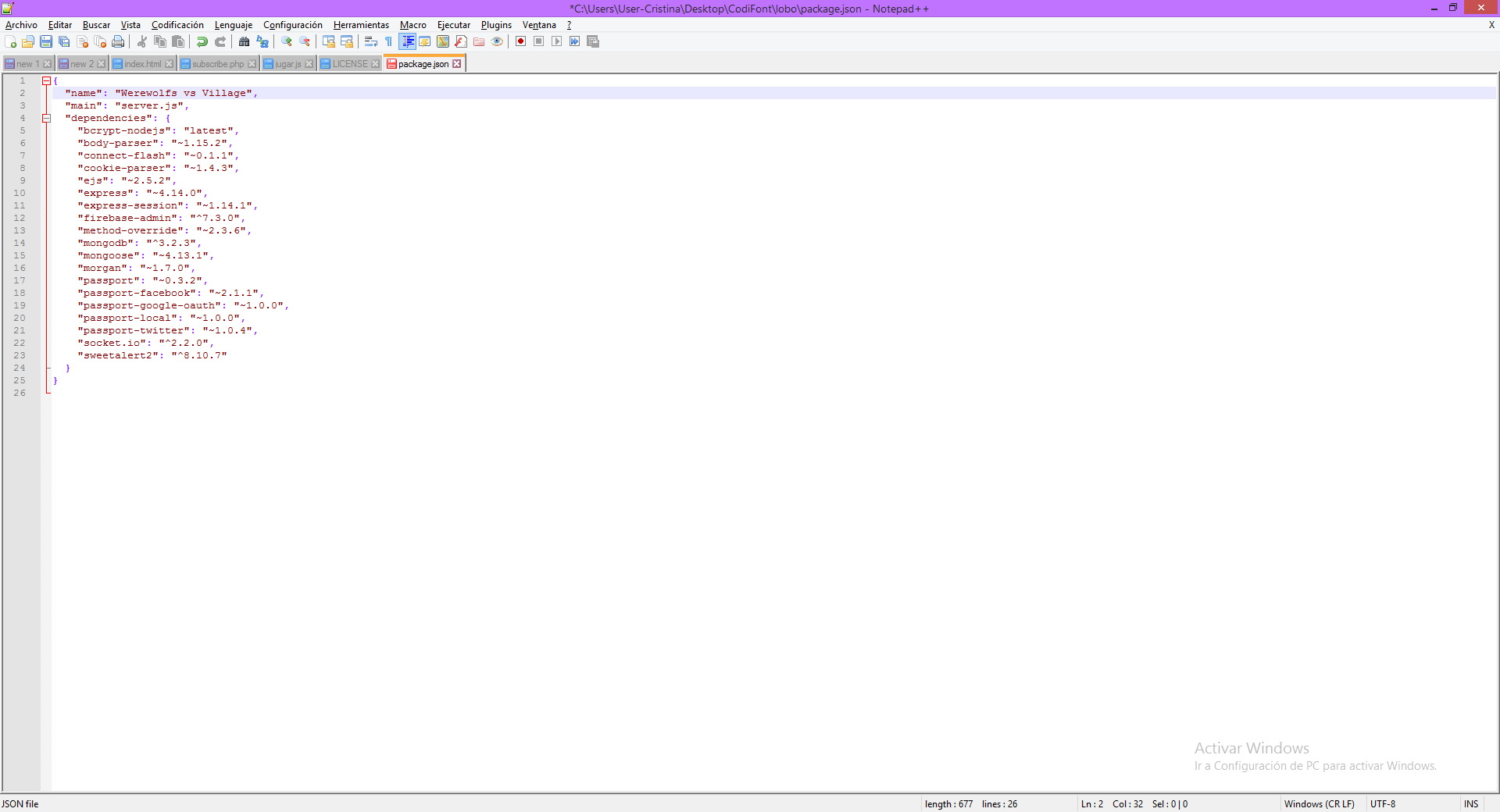
**License**

Llicència de la llibreria PassportJS i la seva estructura web.

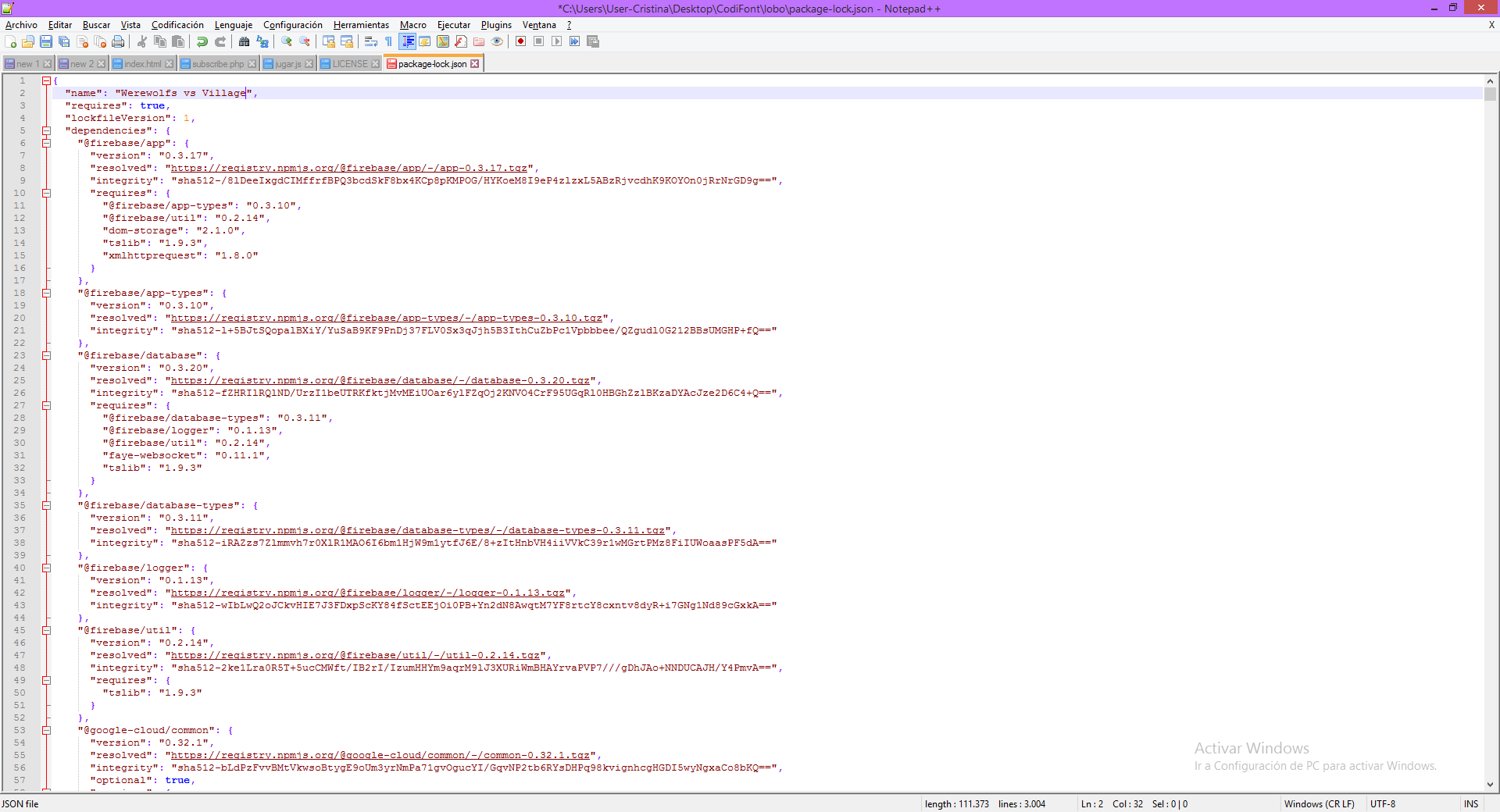


**Package.json**

Informació json de l’aplicació NodeJS

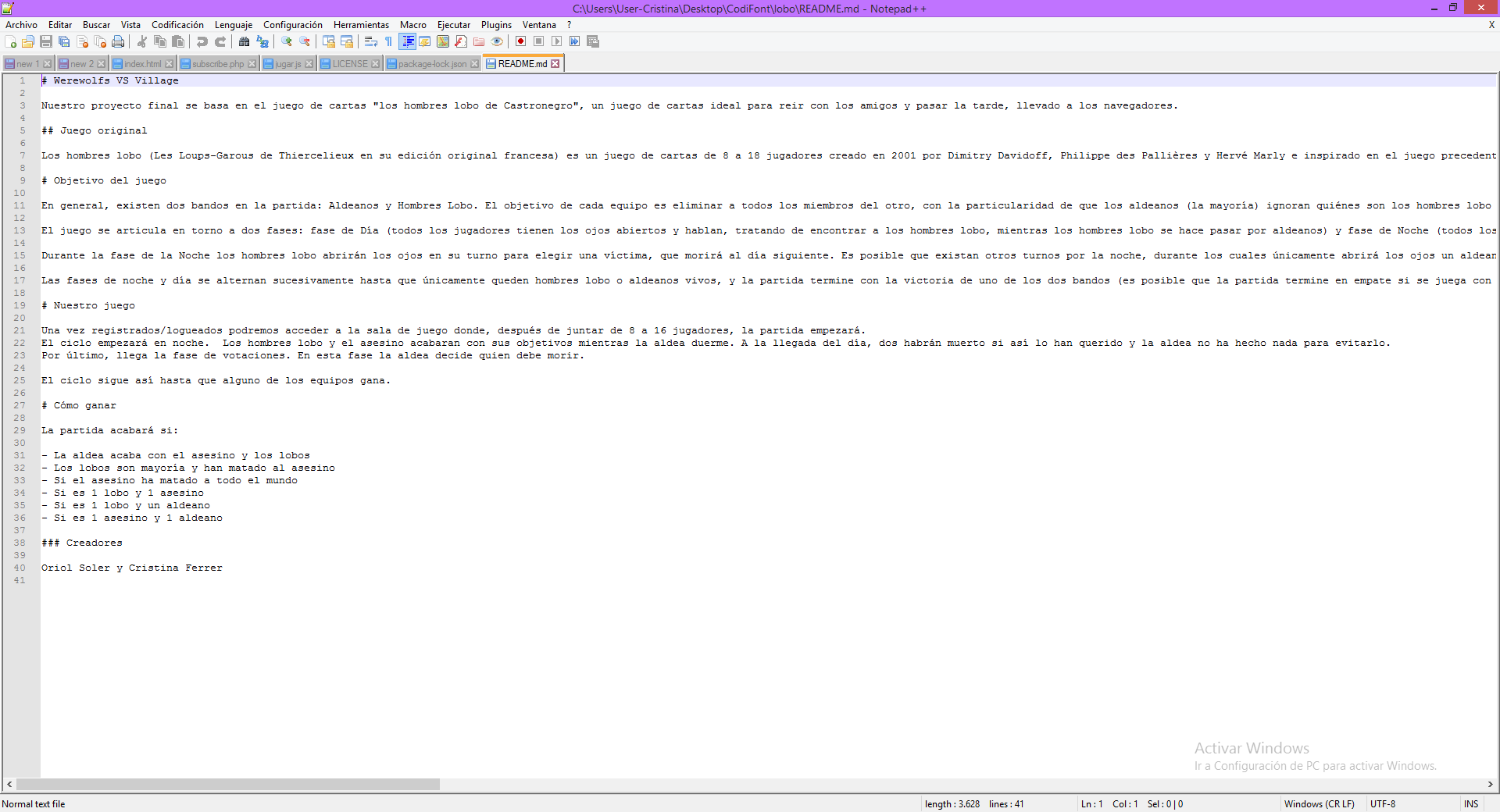


**Package-lock.json**

Fitxer amb la informació de les llibreries que utilitza el NodeJS

**Readme.md**

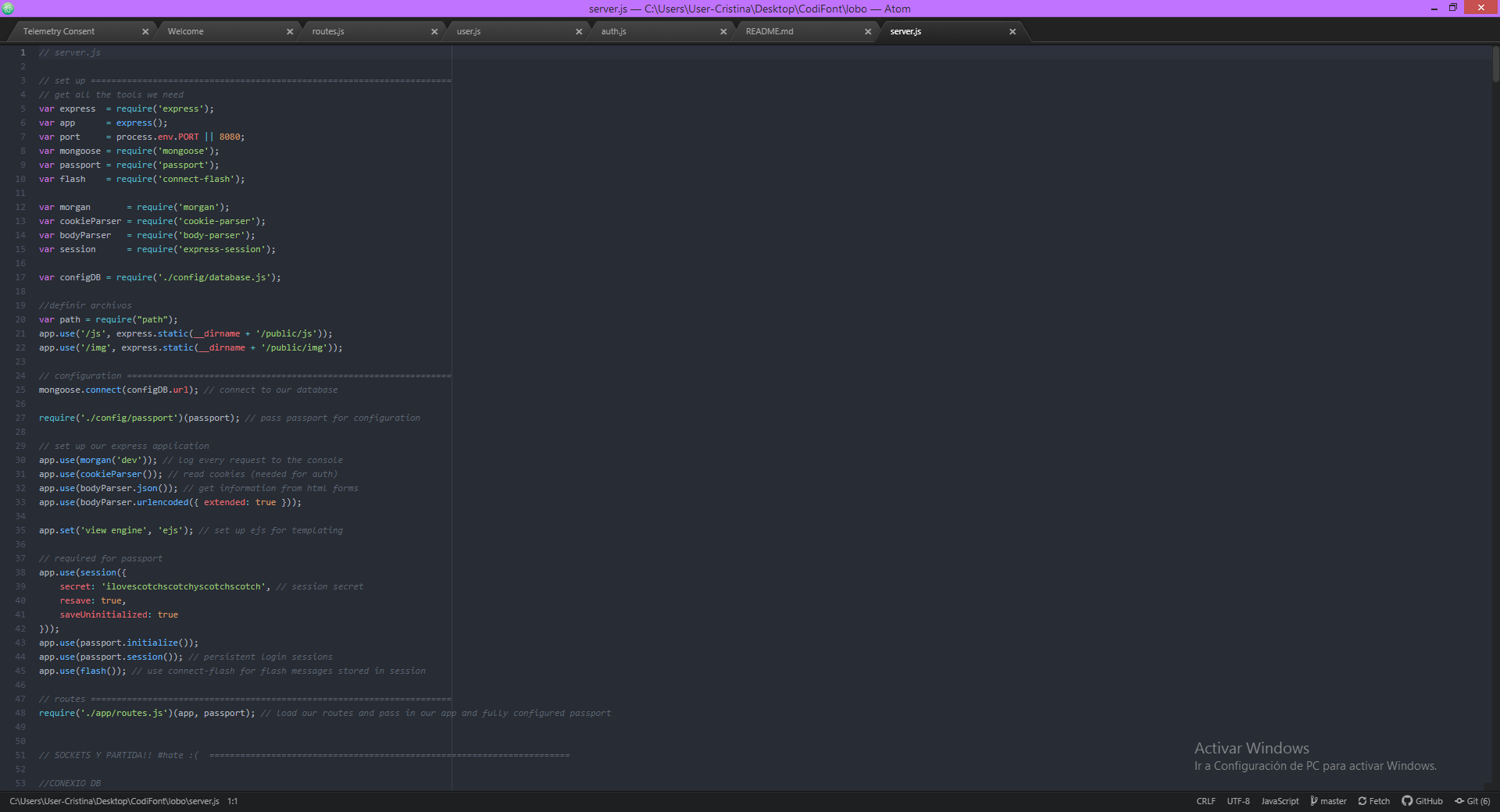
Readme amb els textos de que es el “Lobo” amb els que ens hem basat, com es el nostre joc, com es desenvolupa una partida, com guanyar i els nostres noms.

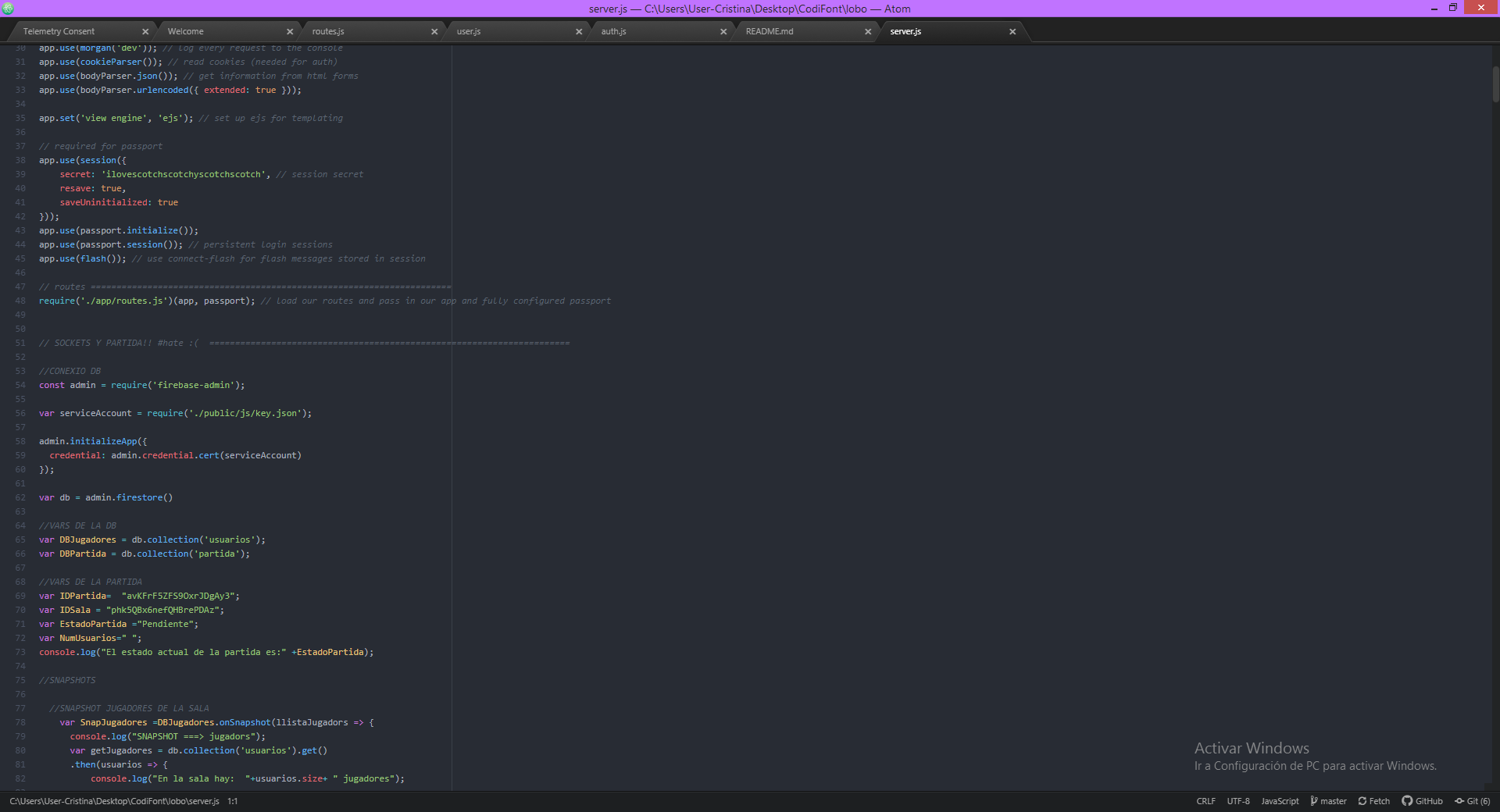


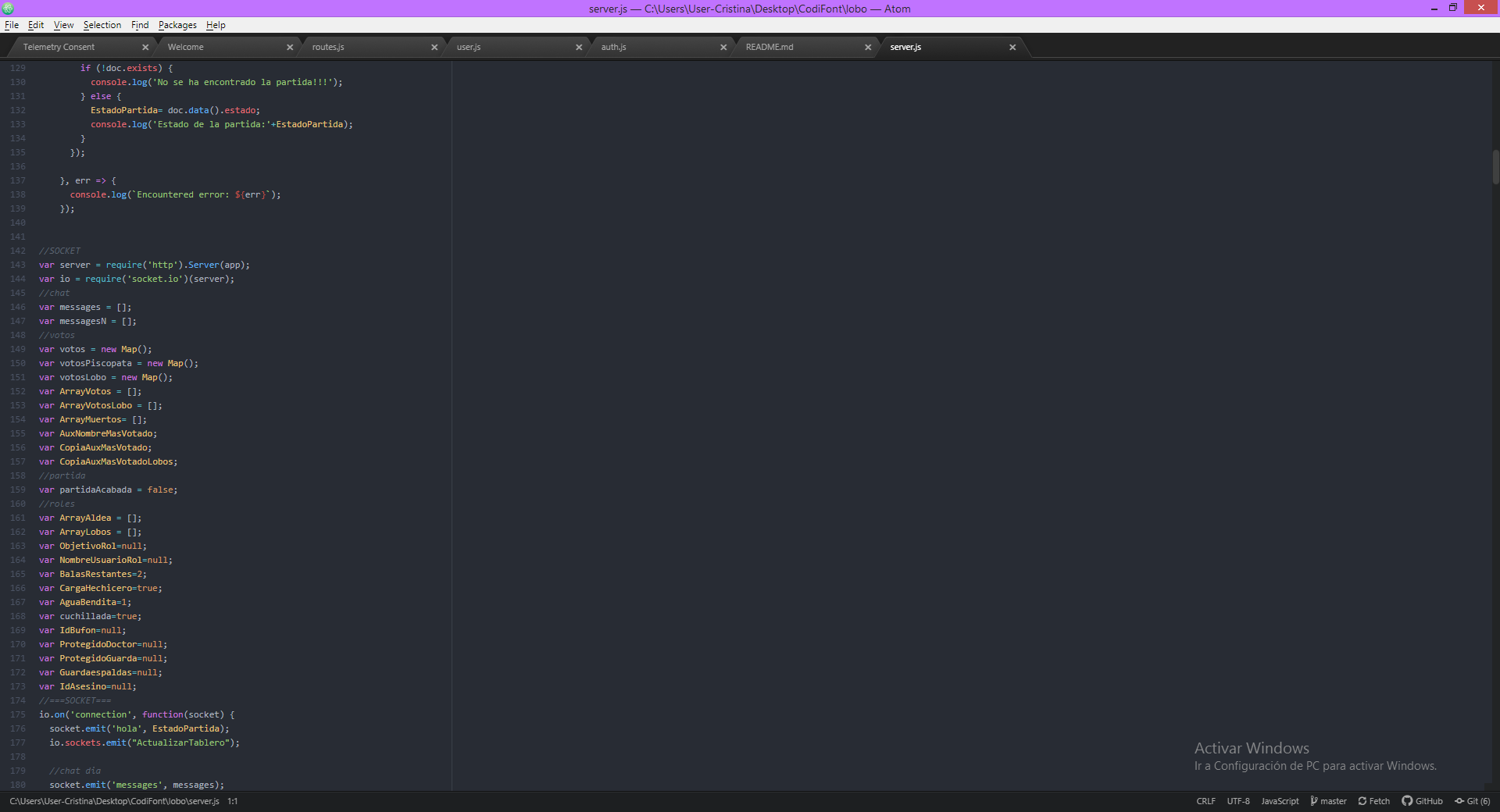
**Server.js**

Arxiu base de la nostra web i de la part servidor de la partida.

Al començament importa totes les llibreries necessàries per funcionar correctament i configura/exporta la configuració del Mongo i el Express.

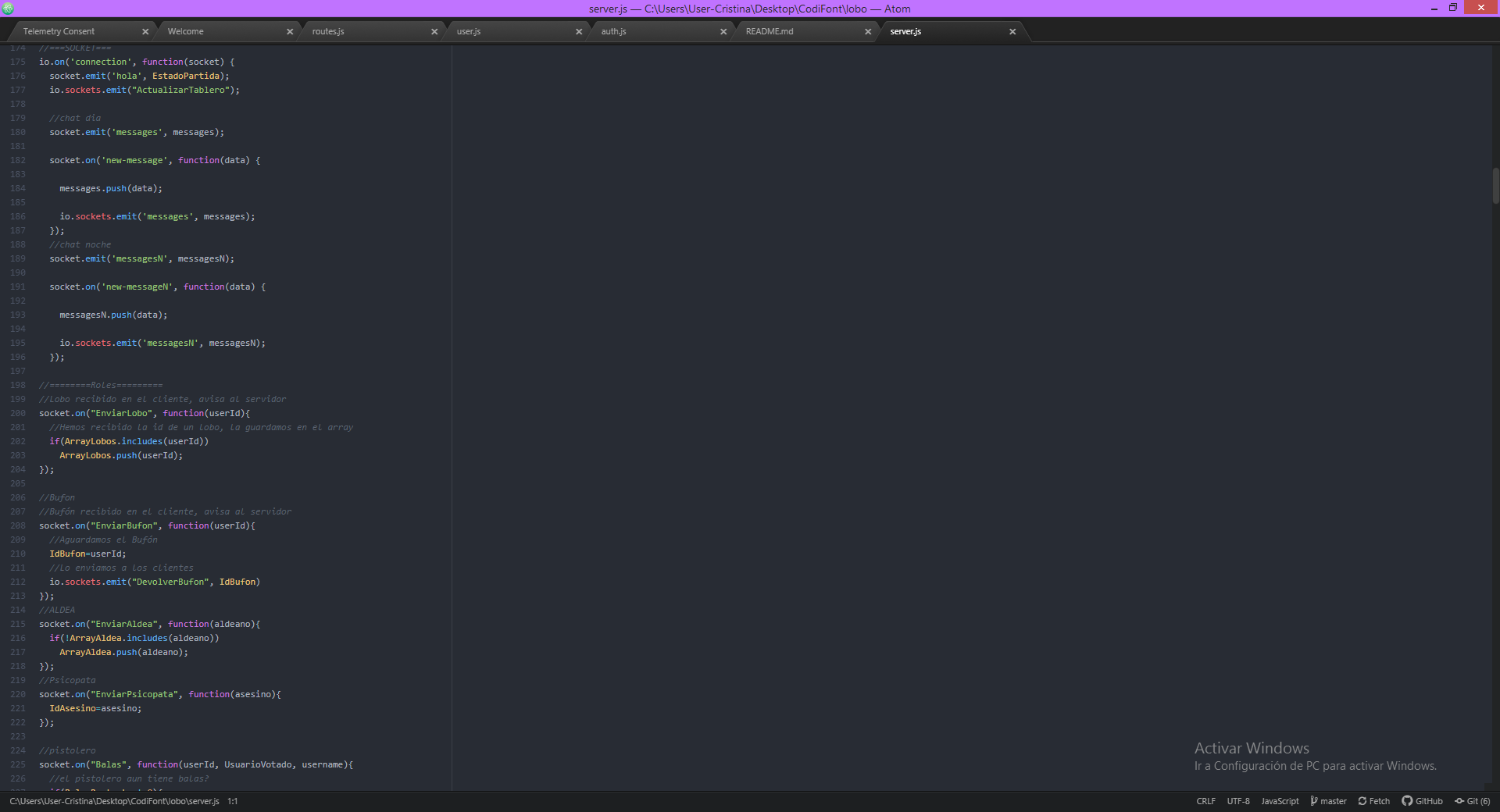
També fa un path directe a la carpeta img i js per poder-ho importar molt més fácil.

Declara les variables necessàries per a la connexió al Firebase, exporta la Key.json personal del projecte, declara variables de la partida...i seguidament controla els snapshots fets a la taula partida i usuarios.

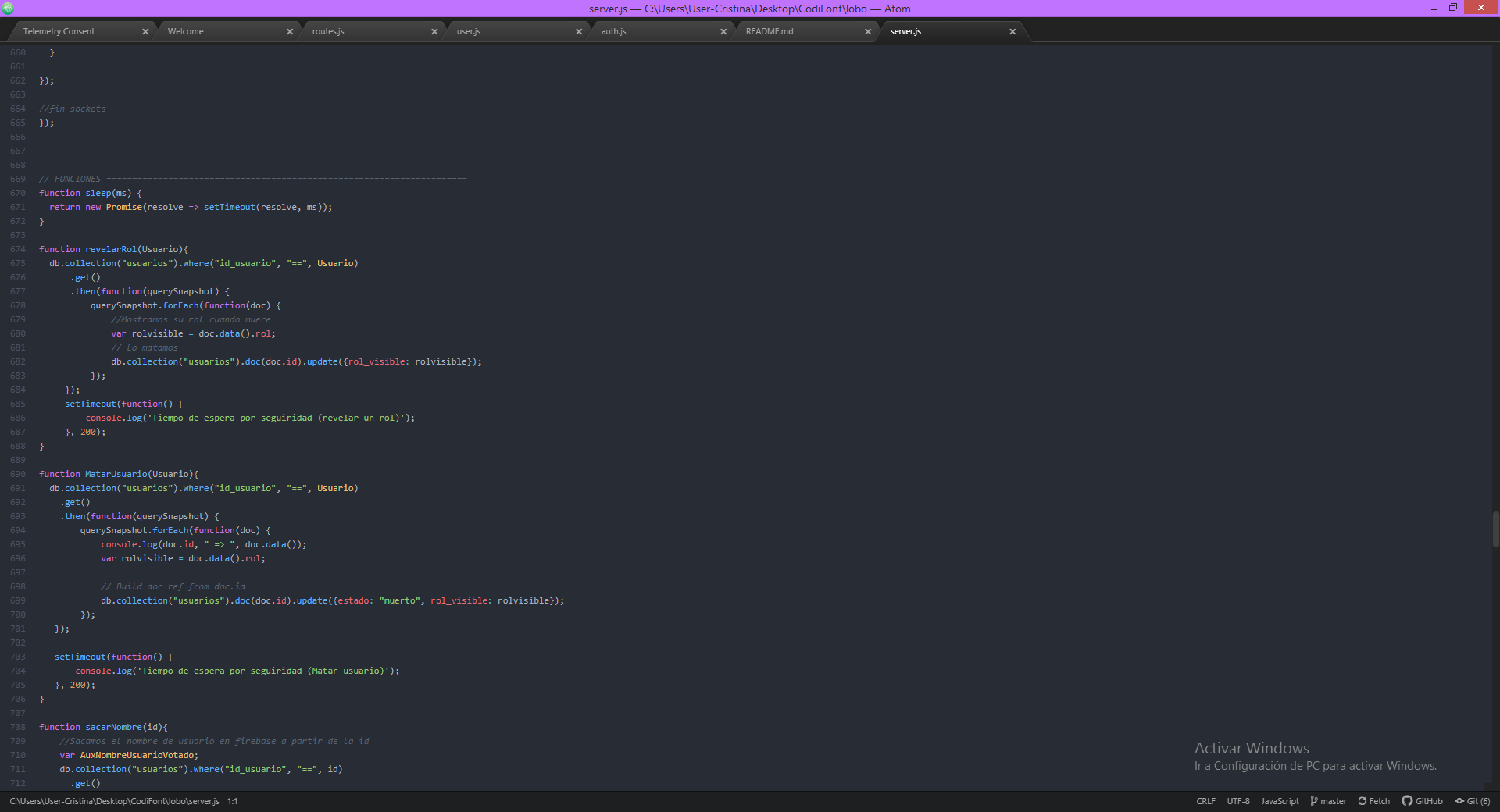


Després d'això, controla els Sockets. Executa les funcions quan rep un Socket o envía una funció amb un emit.

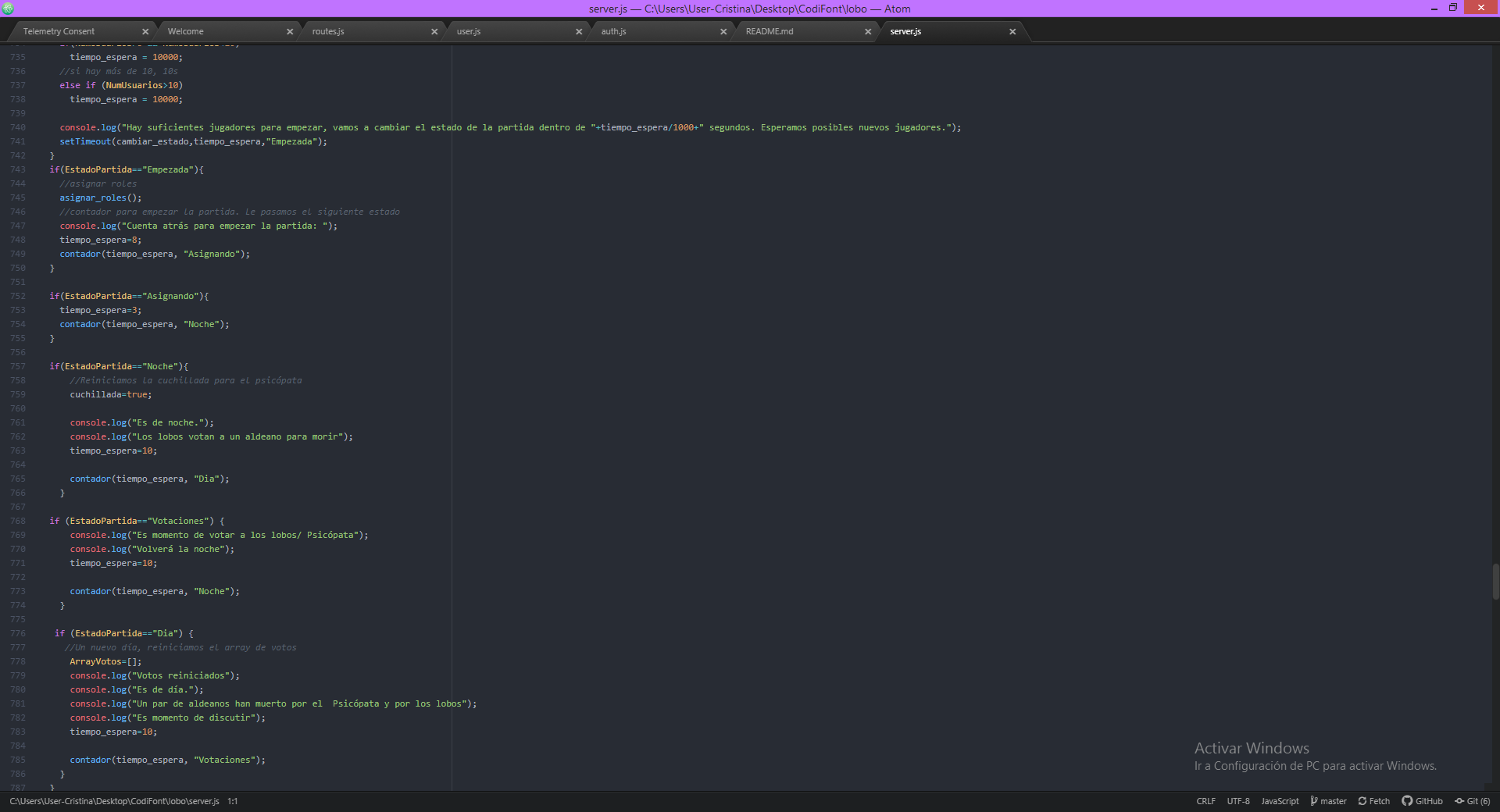
Tots els cicles estan connectats amb Sockets, igual que qualsevol canvi en la partida com la mort d’un usuari o un missatge al xat.



A part de totes les funcions amb sockets també té funcions JS per al funcionament de la partida, funcions per treballar canvis al Firebase o funcions auxiliars (com contadors, esperes de temps, càlculs, etc..)

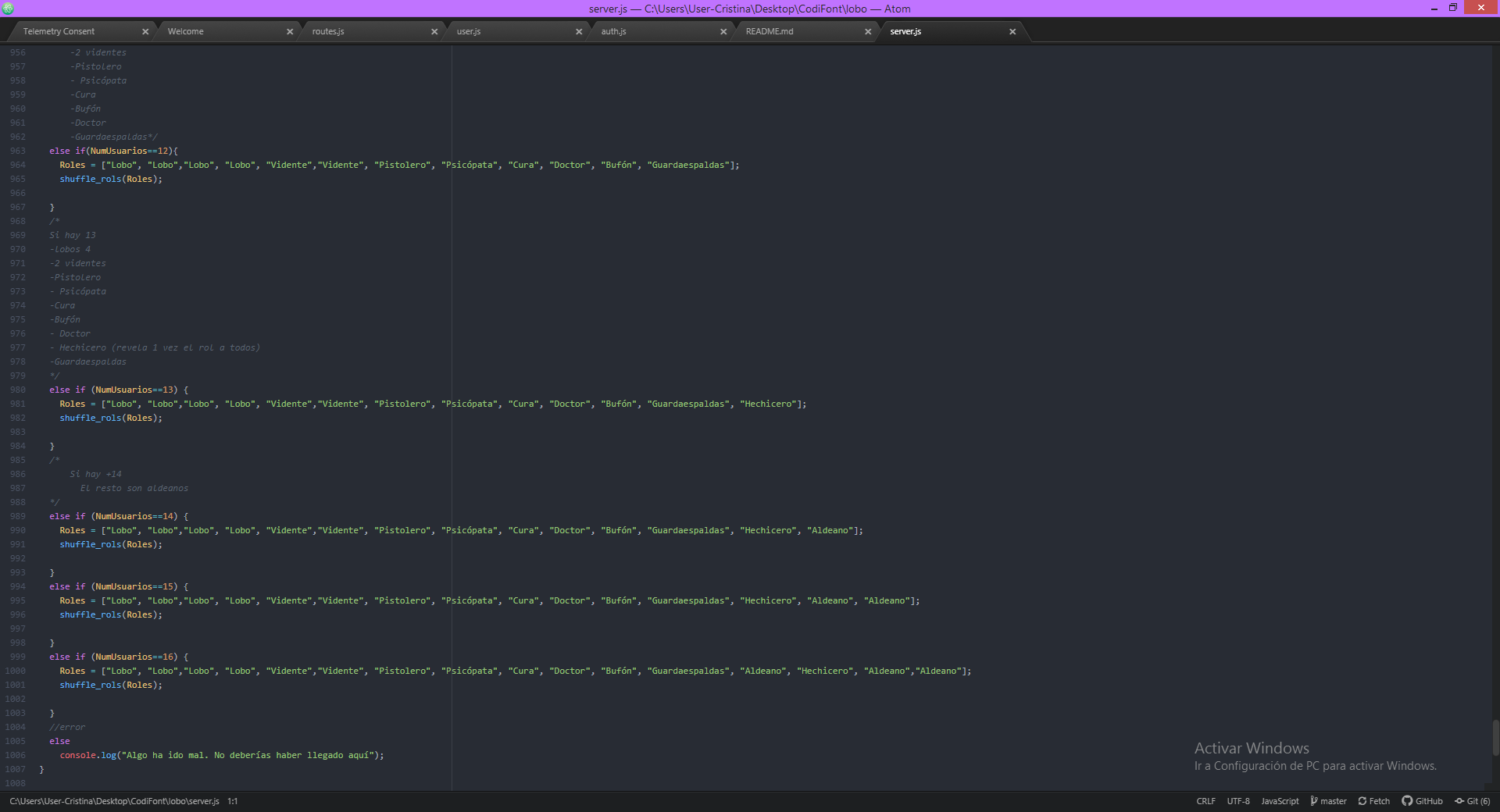


La funció principal (tant per servidor com a client) pel funcionament de la partida, es la funció de “manejar\_estado”. Cada X segons envía un canvi d'estat del següent part del cicle. Després s'encarrega de fer funcionar les funcions de cadascun d’aquest cicles.



Un altre funció important és la funció d’assignar rol de forma aleatoria. Un cop la partida ha començat (s’ha tancat la sala) es començaran a repartir els rols.

Aquesta funció s’encarrega de repartir els valors d’un array o d’un altre segons el número de jugadors per a que cadascun tingui el seu rol. Segons el num de jugadors la partida será d’una manera o d’una altre. No son iguals les partides ni els rols.



Servir la web

