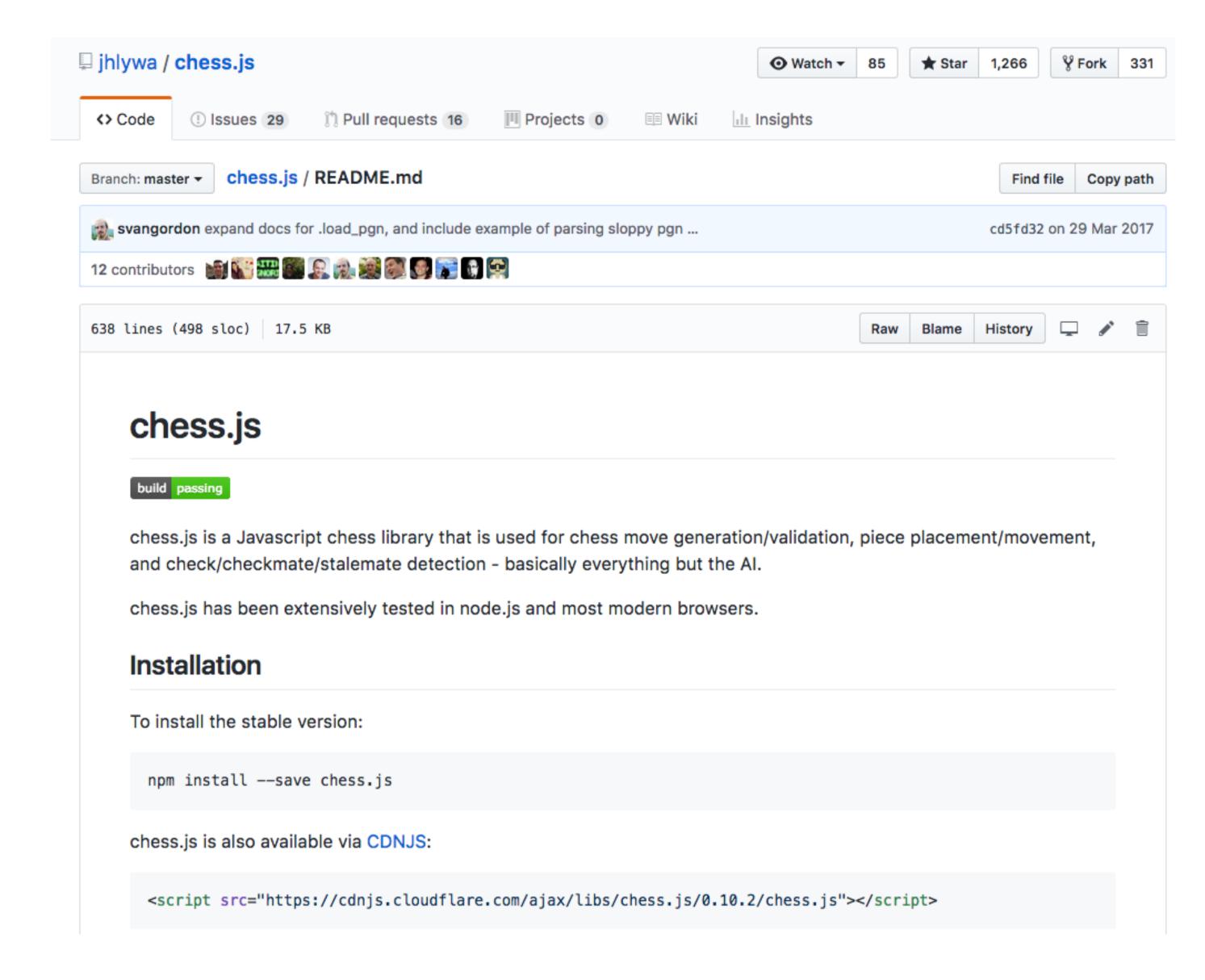
+ Escribe

Personalizar Editar

Estadísticas •••

chess.js

Una vez que hemos preparado<u>una webapp (friendlychat)</u> y tenemos <u>el tablero incrustado en su interfaz de usuario</u>, debemos atacar la cuestión de explicarle a la librería de la inteligencia artificial que en nuestro juego, Fortalezas8x8, únicamente mueven las blancas.



M.X.X.) cada posición tiene un número limitado de movimientos para dar jaque mate. En la modalidad estandar de Fortalezas8x8, por lo general, serán siete. Ver webapp demo: https://fortalezas-

Y, además, habrá que decirle que, por definición (del maestro

```
e42b5.firebaseapp.com/
Ver hoja de pasos del tutorial:
https://github.com/martinsantome/twitter-graph-
visualization/commit/709e09ecf87f0c151b5cc9af3f8235b7c837940e
```

Vamos, entonces, a la tarea. Bienvenido un capítulo más.

Si le echamos un vistazo a la cabecera, en el código de la <u>librería de inteligencia artificial</u> encontramos la declaración de unos objetos. Dado que cuando un programador nombra a una variable trata de hacerlo de un modo tal que quien conozca el nombre con que se bautiza a esa reserva de espacio en memoria para un objeto conozca, además, la utilidad y lógica en el entramado del código. Parece obvio, pero no lo es, ya que hubo un tiempo en que las variables tomaban por nombre escuetas y simples letras del abecedeario. No me extiendo en esta cuestión. Veamos la cabecera:

```
var board = new Array(128);
var kings = {w: EMPTY, b: EMPTY};
var turn = WHITE;
var castling = {w: 0, b: 0};
var ep_square = EMPTY;
var half_moves = 0;
var move_number = 1;
var history = [];
var header = {};
```

con lengua materna distinta. Sin embargo, var board está claro que contiene un tablero de de 128 posiciones. Primero, un apunte técnico que arrastramos de las entregas

Están en inglés. Ya lo lamenta un escritor que tiene audiencia

anteriores. Publicar la webapp en el navegador local, subirla al repositorio:

```
Your branch is ahead of 'origin/fortalezas88' by 2 commits.
  (use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track) aleph@oxs-Mac-mini:~/Desktop/friendlychat/cloud-functions$ firebase serve
=== Serving from '/Users/aleph/Desktop/friendlychat/cloud-functions'...
i functions: Preparing to emulate functions.
i hosting: Serving hosting files from: ./
hosting: Local server: http://localhost:5000
Warning: You're using Node.js v8.9.1 but Google Cloud Functions only supports v6.11.5.

i functions: No HTTPS functions found. Use firebase functions:shell if you would like to emulate other types of functions.
i functions: No HTTPS functions found. Use firebase functions: shell if you would like to emulate other types of functions.
^CShutting down...
aleph@oxs-Mac-mini:~/Desktop/friendlychat/cloud-functions$ git push -u origin fortalezas88
Counting objects: 26, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (26/26), 80.54 KiB | 0 bytes/s, done.
Total 26 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/martinsantome/twitter-graph-visualization.git
   709e09e..4b60cd3 fortalezas88 -> fortalezas88
Branch fortalezas88 set up to track remote branch fortalezas88 from origin. aleph@oxs-Mac-mini:~/Desktop/friendlychat/cloud-functions$
```

SALIDA CONSOLA DE DEPURACIÓN TERMINAL PROBLEMAS Delta compression using up to 4 threads.

Y, si además, queremos publicar en la nube:

```
Compressing objects: 100% (24/24), done.
Writing objects: 100% (26/26), 80.54 KiB | 0 bytes/s, done.
Total 26 (delta 1), reused 0 (delta 0)
   remote: Resolving deltas: 100% (1/1), done.
   To https://github.com/martinsantome/twitter-graph-visualization.git
          709e09e..4b60cd3 fortalezas88 -> fortalezas88
   Branch fortalezas88 set up to track remote branch fortalezas88 from origin.
   (reverse-i-search)`de': git add index.html
aleph@oxs-Mac-mini:~/Desktop/friendlychat/cloud-functions$ firebase deploy
     === Deploying to 'fortalezas-e42b5'...
   i deploying database, storage, functions, hosting
   i database: checking rules syntax...
i database: checking rules syntax...
/ database: rules syntax for database fortalezas-e42b5 is valid
i storage: checking storage.rules for compilation errors...
/ storage: rules file storage.rules compiled successfully
i functions: ensuring necessary APIs are enabled...
/ functions: all necessary APIs are enabled
i storage: uploading rules storage.rules...
i functions: preparing functions directory for uploading...
i functions: packaged functions (40.75 KB) for uploading
/ functions: functions folder uploaded successfully
i hosting: preparing ./ directory for upload...
/ hosting: 10092 files uploaded successfully
i database: releasing rules...
   i database: releasing rules...
 database: releasing rules...
database: rules for database fortalezas-e42b5 released successfully
storage: released rules storage.rules to firebase.storage/fortalezas-e42b5.appspot.com
i functions: creating function addWelcomeMessages...
i functions: creating function sendNotifications...
functions[addWelcomeMessages]: Successful create operation.
functions[sendNotifications]: Successful create operation.
     Deploy complete!
  Project Console: https://console.firebase.google.com/project/fortalezas-e42b5/overview
Hosting URL: https://fortalezas-e42b5.firebaseapp.com
aleph@oxs-Mac-mini:~/Desktop/friendlychat/cloud-functions$
```

Entonces, apartándonos de estos temas, digamos, un poco más de tipo administrativo, centrándonos en la inteligencia artificial.

NOTA: SE REGALA ESTE SITE

Una de las librearías más simples pero robustas halladas a tal efecto ha sido: Chess.js valoración:

Conclusión: Una librería indispensable en cualquier caja

Pull requests Issues Marketplace Explore

+ + = == -

```
Nivel de dificultad: moderado
Interés: moderado
```

utf8=%E2%9C%93&q=turn+%3D

blancas para dar mate.

jhlywa/chess.js turn =

Útil: del todo y máximo.

de herramientas ludita. Puntos claves: https://github.com/jhlywa/chess.js/search?



Ya se puede jugar en el sitio demo. Requiere una cuenta gmail. Nota: no usar el chat que todavía está en proceso de implantación. Veamos una posible (no la ganadora) solución:

mínimo de movimientos consecutivos que habrán de hacer las

BLANCAS: Jaime S NEGRAS: nadie (fortaleza formal) FECHA: sábado 12 de mayo, sesión de tarde

```
LUGAR: acebuchesLab
1. e1, -
2. Cf3, -
3. Cg5, -
4. Df3!!, -
5. Df7++
```



← Thinkers vs Artificers

```
Responder
 Introduce aquí tu comentario...
```

javascript, matrices 8x8

Editar

```
Buscar ...
                                   Archivos
                                    • <u>junio 2019</u>
```

mayo 2019