



Proyecto 2ª Evaluación

Joel René López Falcón
Carlos Sánchez Rodríguez

Componentes en Angular para unificar diferentes
visualizadores relacionados con la bioinformática.

Índice

Introducción	3
Gestión de datos	3
Backend	3
Frontend	3
Metodología de trabajo	6
Repositorios	6
Comparación de tecnologías	6
Conclusiones	6
Referencias	7

1. Introducción

Este documento se aporta como parte de la documentación de una aplicación principalmente para dispositivos móviles (siguiendo el principio de mobile-first) que desarrollará su función principalmente entorno a una integración directa de distintos componentes que permitan la visualización de secuencias filogenéticas ya sea en forma de alineamiento o árbol.

El diseño seguirá las normas de “material design”, con sus estándares y siguiendo una línea para fomentar el uso responsivo. (Multiplataforma)

La estructuración de la aplicación consta de:

- Proyecto en Angular (JavaScript) con datasets escogidos de distintas fuentes.

En este caso es totalmente frontend, lo que quiere decir que no necesita de un servidor (o backend) para su funcionamiento completo.

2. Gestión de datos

Como se ha mencionado anteriormente, esta aplicación es totalmente frontend así que en cuanto a la gestión de datos no hay mucho que detallar más allá de la posibilidad para el usuario de abrir unos ficheros desde las propias herramientas de visualización que traen integradas estas funciones.

El formato de estos ficheros va dependiendo del caso, pero hemos encontrado los diferentes como los más comunes: .fasta, .aln o. json

3. Backend

Desconocemos hasta el momento si se va a hacer un uso de un servidor o “backend” para la obtención de los datos o para todo el ecosistema general de la app en un futuro.

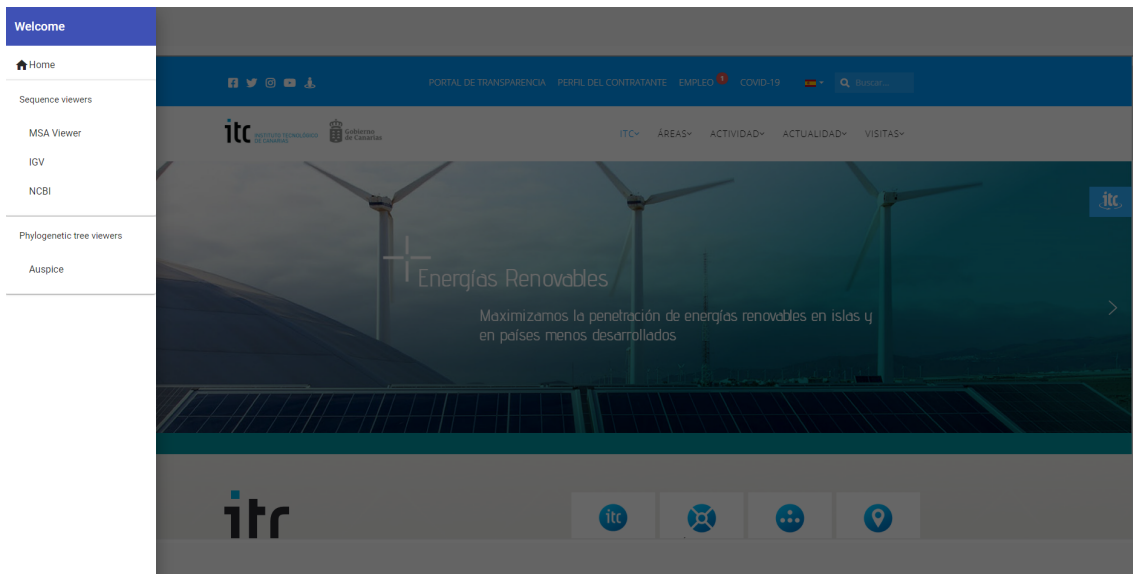
4. Frontend

En la introducción ya se informaba de que la aplicación enteramente está focalizada en un cliente o “frontend”.

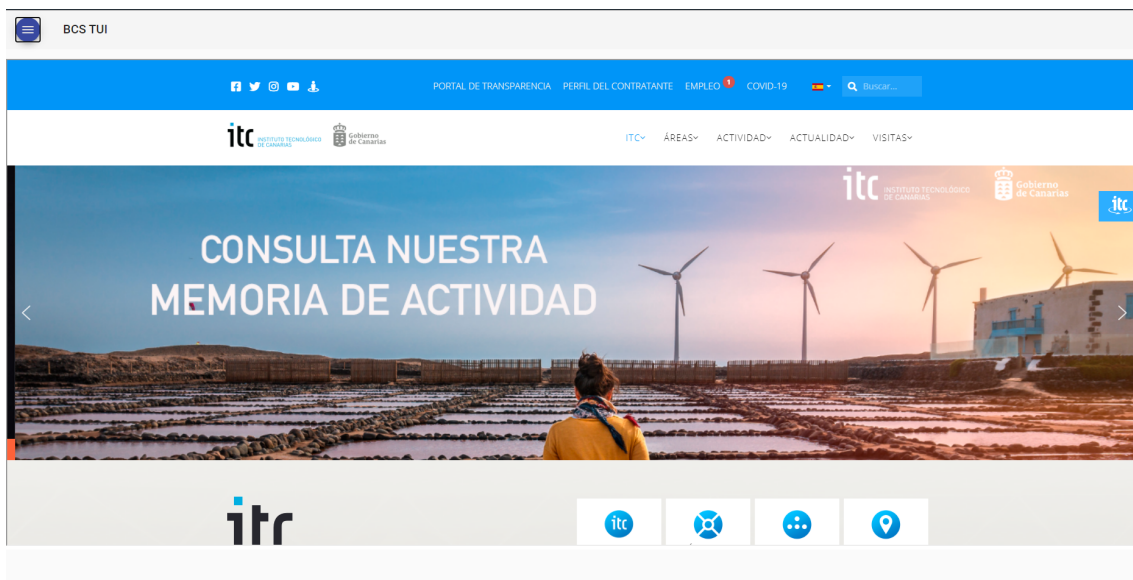
En esta parte se nos pidió una aplicación basada en Angular capaz de contener diferentes componentes que encapsulan herramientas de bioinformática para el análisis de diferentes datos tal y como se ha detallado anteriormente.

La parte previa al desarrollo de esta aplicación ha constado de un planteamiento del diseño y la idea.

El cliente concretamente quería una barra de navegación lateral con una categorización de los diferentes visualizadores. Aspecto que en este caso ha sido totalmente cubierto como se muestra en la siguiente captura:

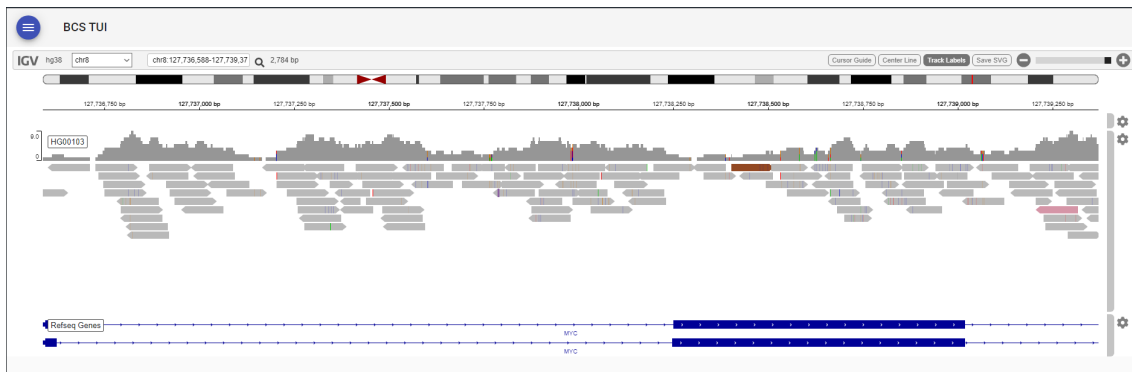


Como página de inicio para la aplicación en desarrollo hemos visto conveniente la incorporación de la página principal del ITC ya que nos resultaba bastante útil poder disponer de todo lo necesario como consultar información relativa al cliente en este caso directamente desde dentro del proyecto una vez ejecutándose. Cabe destacar que lo que se busca son realmente extraer los componentes desarrollados y no la aplicación al completo.



Los visualizadores están categorizados en sequence viewers o phylogenetic tree viewers (visualizadores de secuencias o visualizadores de árboles filogenéticos)

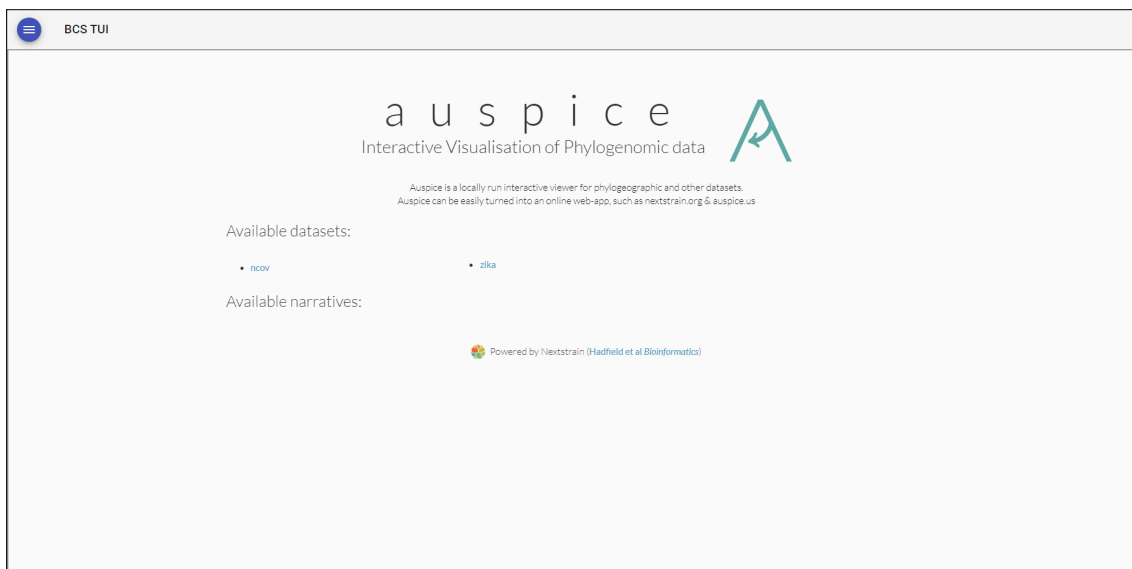
En el caso de las secuencias, el que mejor hemos visto ha sido el IGV por sus completos apartados y la multitud de opciones que trae integradas con tan solo instalar el paquete de node que se encuentra en su repositorio [oficial](#).



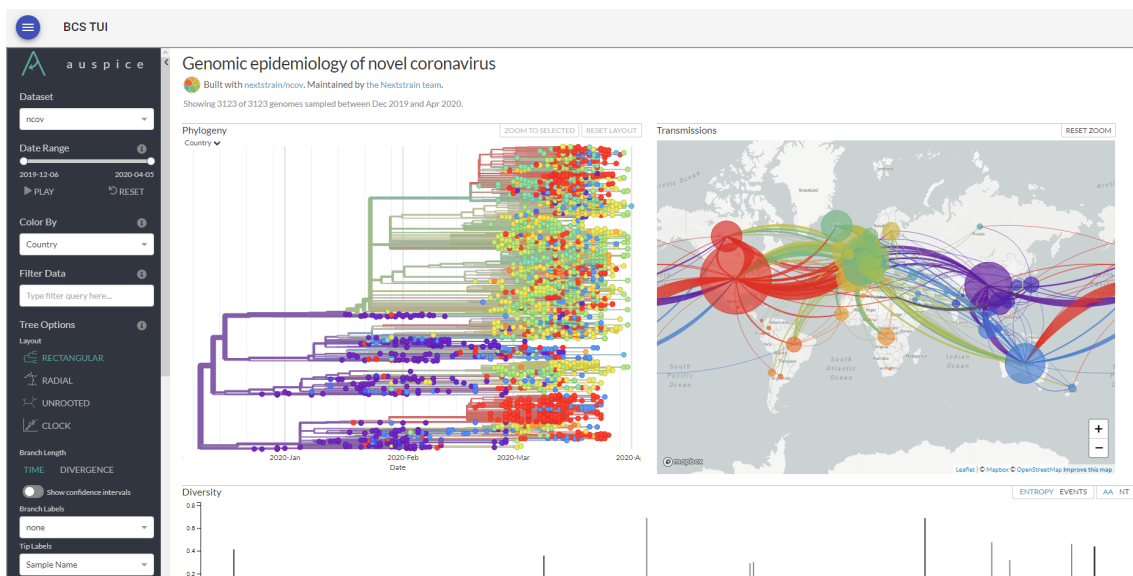
Y, por otra parte, en las vistas de árbol, actualmente contamos con únicamente un visualizador integrado, pero viendo el resto de opciones que planteaba el cliente, creemos que es también, Auspice, el más completo entre ellos.

El único inconveniente, si es que se puede tener en cuenta como tal, es que requiere del inicio de un cliente externo que hasta el momento no hemos sido capaces de automatizar dentro del funcionamiento del núcleo de Angular.

Al iniciar dicho cliente por otra parte dispondremos de una vista encapsulada como esta, que no es más que una integración desde la web. Como podemos observar, existe un menú que nos permitirá seleccionar el dataset a utilizar.

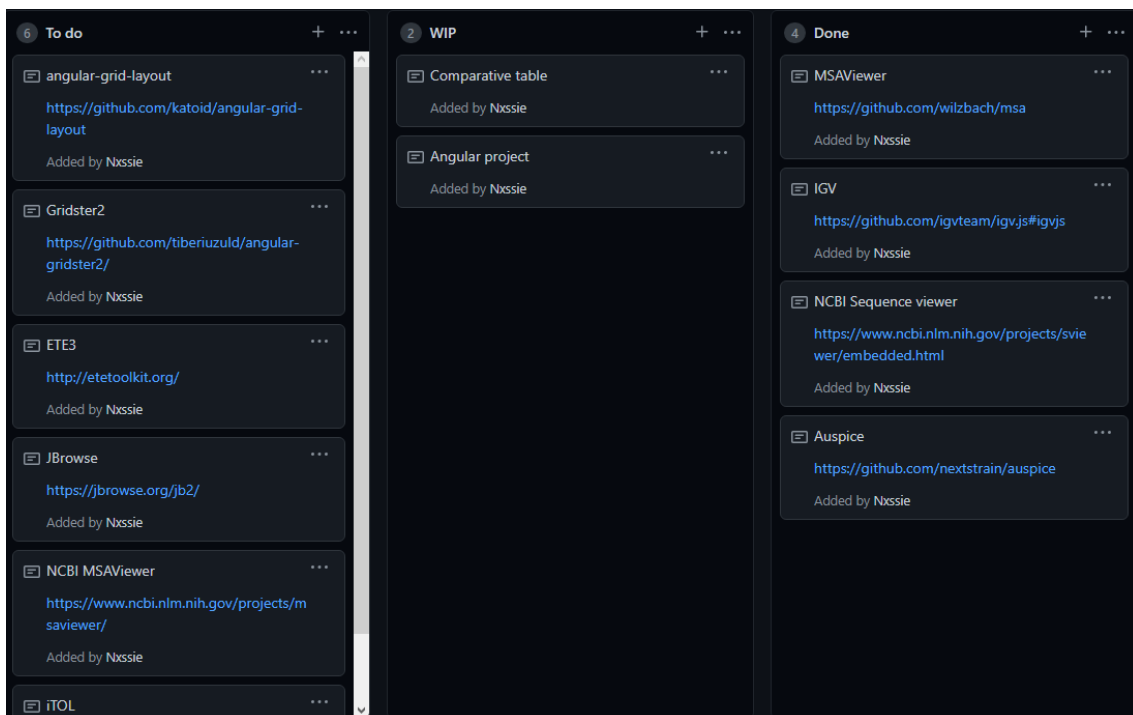


Y ya con el dataset elegido, veremos la herramienta en sí:



5. Metodología de trabajo

Nos hemos dividido y gestionado las diferentes tareas referentes al proyecto mediante el uso de un excel facilitado por la propia empresa para utilizar como kanban, dado que este excel es privado también hemos reflejado las tareas en el gestor de proyectos del propio GitHub.



6. Repositorios

Este proyecto se ha trabajado principalmente en un repositorio privado facilitado por la empresa, sin embargo la última semana de trabajo se ha llevado a cabo en el repositorio en el que se presenta este documento, para poder tener un repositorio [público](#) con el que presentar el proyecto.

7. Comparación de tecnologías

Las principales tecnologías empleadas para esta aplicación son las herramientas de visualización, ya que Angular tan solo sirve como esqueleto para poder acceder a todas desde un mismo lugar. Si comparamos cada componente:

- MSA: El visor más simple, ya que permite únicamente ver las cadenas genéticas del archivo que le pidas, sin embargo trae una buena cantidad de opciones, como importar archivos desde múltiples fuentes o configurar la interfaz.
- IGV: El visor más completo, permite no solo ver las cadenas del archivo que hayas introducido, sino además consultar diferentes anotaciones que haya en el mismo archivo, entre otras múltiples opciones que presenta.
- NCBI: Presenta opciones similares a las del IGV, sin embargo este visor no es un componente como tal, sino que está embebido de su página web original, lo que resultaría en problemas a la hora de trabajar sin conexión.
- Auspice: El único visor de árboles integrado en esta versión del proyecto, sin embargo se ha integrado este en concreto tras valorar las opciones que ofrecía en un primer vistazo en comparación con el ETE3 y el iTOL. Auspice ofrece una increíble cantidad de opciones para modificar la interfaz en la que vemos los árboles, además de ser sencillo de instalar y ejecutar. Sin embargo tiene los contratiempos de que es necesario ser ejecutado como un servidor independiente de la aplicación original y de que tan solo lee archivos en formato .json.

8. Conclusiones

Al tratarse de un apartado más personal y bajo el punto de vista de nuestro trabajo como compañeros debe tomarse como tal y de ninguna manera supone una vista experimentada u objetiva de todo lo detallado anteriormente.

Creo que ambos hemos visto el potencial que pueden tener estas herramientas en el uso cotidiano de un profesional que se dedique a diario al campo de la bioinformática. Aunque sí es cierto que todavía, desde el desarrollo de estos componentes creemos que le queda mucho rodaje para que sean completamente perfectas, un ejemplo de ello ha sido las múltiples ocasiones en las que nos hemos chocado de frente con una documentación un tanto dejada y abstracta que a veces resultaba difícil de entender para el usuario que, a fin de cuentas, iba a implementar dicha herramienta. Algo que creemos, es primordial corregir para un mejor flujo de trabajo.

También notamos una falta de estandarización en cuanto la implementación de algo tan concreto como son los diferentes archivos que se utilizan. No vemos factible que dos programas que se usan con un mismo fin, sean tan diferentes y usen

denominaciones tan dispares entre sí para referirse a un mismo hecho u objeto a tratar.

9. Referencias

- <https://github.com/wilzbach/msa>
- <https://github.com/igvteam/igv.js#igvjs>
- <https://www.ncbi.nlm.nih.gov/projects/sviewer/embedded.html>
- <https://github.com/nextstrain/auspice>