



BCnc

CONSULTING GROUP

### **CÓMO LEER EL DOCUMENTO:**

- Este documento tiene como objeto servir como guía de apoyo para realizar el test proporcionado.
- Se presentan los criterios de evaluación.
- Ante cualquier duda, por favor contactad con vuestro contacto de HR en BCNC y os solventarán las dudas pertinentes.

# FRONT-END TEST GUIDANCE

## CRITERIOS DE EVALUACIÓN DE LA PRUEBA:

En esta sección queremos compartir los criterios de evaluación que se seguirán para evaluar la prueba técnica, puede que algunos de ellos no estén detallados en la propia prueba, pero entendemos que se tienen en cuenta y por lo tanto queremos haceros conocedores/as de ellos:

1. Un buen readme explicando tecnologías, arquitecturas y patrones utilizados, ejemplo de los tests, además de cosas que debe tener un readme (es una muy buena oportunidad y se valora altamente la explicación de las decisiones de diseño tomadas, ya que esto ayuda a comprender la lógica que has seguido para la toma de decisiones, pudiendo remarcar y demostrar así el seniority sobre la materia):
2. Claridad del código, se tiene en cuenta lo fácil que resulte leerlo y entenderlo (utilizando principios SOLID):
3. Número de alertas que tenga cuando se ejecute un linter y que tenga un formatter:
4. Separación en capas y que cada capa contenga únicamente su responsabilidad, explicar en el readme la arquitectura elegida y el porqué). (importante que el diseño sea escalable y modular):
5. Consola limpia de errores:
6. El diseño es responsive y se hace uso desde cero de CSS (para demostrar el dominio del mismo):
7. Uso de las herramientas que proporciona Github o similares para la gestión del código (utilizar nomenclatura estándar de commits, ramas, etc):
8. Que pasen correctamente los tests para ver el desarrollo del mismo:
9. Que la solución sea modular, basada en componentes (explicar en el readme el porqué se ha tomado la decisión de dividir en esos componentes):
10. Uso de Webpack o similar para la solución al completo:
11. Implementación de custom hooks:
12. Uso de Typescript con tipado completo en toda la aplicación:

**Continúa en la siguiente página...**

### CRITERIOS DE EVALUACIÓN DE LA PRUEBA:

13. Se utilizan componentes from scratch:
14. Uso de caché para aumentar la eficiencia (explicar en el readme qué mejoras se consiguen y la estrategia a seguir):
15. Diseño fiel al propuesto:
16. Modo desarrollo en el que se sirvan los assets sin minimizar (pueden estar concatenados) y otro modo producción donde se deben servir los assets concatenados y minimizados:
17. Cumple con todos los requisitos pedidos en los apartados de "Funcionalidad" de la prueba (sí/no por qué?):
18. Cumple con los requisitos de:
  - No usar librerías de componentes como antd, reactstrap, materialui etc...
  - Los componentes deben ser creados desde cero por ti mismo.
  - Para la gestión de estado, usar ContextAPI de React
19. ¿Se han aplicado las siguientes mejoras?:
  - La aplicación puede estar desplegada:
  - Uso de SSR (Es posible usar Next.js):
  - Se valorará el uso de variables CSS:
20. Que el ejercicio se pueda probar y cumpla exactamente con lo que se pide "Requerimientos del enunciado" (se puede extender funcionalidad, pero nunca debe comprometer o modificar el funcionamiento principal):