**PRUEBA TÉCNICA PARA EL CARGO DE DESARROLLADOR DE SOFTWARE**

|  |  |
| --- | --- |
| **Tiempo** | Sin límites.  **Nota**: El tiempo que usted invierta en la prueba será un ítem evaluado. |
| **Tecnologías** | **Frontend**: Angular 10+  **Backend**: (Laravel o .Net o Node js o Deno js)  **Nota**: Usted decide si hacer la prueba fullstack (Frontend y Backend), cada entorno es un ítem evaluado. |
| **Criterios técnicos** | * Clean code * Clean Architecture * Patrones de diseño * Principios solid * Docker y Docker-compose * POO * Pruebas unitarias * Git * Protocolos de conexión e integración. * Microservicios * Manejo de estilos (CSS, Bootstrap, PrimeNG, Angular material) * Persistencia de datos SQL. * Server-side.   **Nota**: No es obligatorio tener en cuenta todos los criterios, pero cada uno será un ítem evaluado. |
| **Criterios de contenido** | * Login. * Atributos(Crear, Leer, Actualizar, Eliminar y listar) * Producto(Crear, Leer, Actualizar, Eliminar y listar). * Navbar. * Footer.   **Nota**: No es obligatorio tener en cuenta todos los criterios, pero cada uno será un ítem evaluado. |
| **Descripción del problema** | Se debe desarrollar una web app, el cual permita, con un inicio de sesión administrado, acceder al componente productos y atributos de calidad de Pervolare.  El componente atributos debe permitir crear, leer, actualizar, eliminar (suave o lógico) y listar; Los atributos pueden ser los colores del producto, las tallas, las marcas, etc.  El componente de productos debe permitir crear, leer, actualizar, eliminar y listar; dentro del leer se debe poder agregar los atributos que se deseen al producto, estos deben combinarse entre sí y poder listarse dentro de la misma vista (**No invierta tiempo haciendo crud, solo listar**), ejemplo:   |  |  |  | | --- | --- | --- | | Producto: Camisa | Agregar atributo: color blanco | Resultado: Camisa color blanco | | Producto: Camisa | Agregar atributo: talla xs | Resultado: Camisa color blanco - talla xs | | Producto: Camisa | Agregar atributo: color azul | Resultado: Camisa color blanco - talla xs  Camisa color azul - talla xs | | Producto: Camisa | Agregar atributo: talla s | Resultado:  Camisa color blanco - talla xs  Camisa color azul - talla xs  Camisa color blanco - talla s  Camisa color azul - talla s | | Producto: Camisa | Agregar atributo: marca gucci | Resultado: Camisa color blanco - talla s - marca gucci  Camisa color azul - talla s - marca gucci  Camisa color blanco - talla xs - marca gucci  Camisa color azul - talla xs - marca gucci |   **Nota: el resultado es la lista que debemos ver en pantalla una vez se agregue el resultado** |
| **Interfaces** | **Login**:   * email * password   **Producto**:   * id * name * value * description * createDate * updateDate * softDelete   **Atributo:**   * id * name * type (Color, Talla, Marca, Fábrica) * createDate * updateDate * softDelete   **Nota: En caso de ser necesario puede generar interfaces, tablas pivote o auxiliares** |
| **Lógica de negocio** | **Login:**   * **email:** Verificar que sea un email válido. * **password:** validar que sea una contraseña de 8 a 16 caracteres alfanuméricos, permitiendo solo “\*/+-$%&” como caracteres especiales.   **Producto**:   * **id:** Un identificador único universal. * **name:** Validar que sea un título de 2 a 10 caracteres alfanuméricos. * **value**: caracteres numéricos > a 0. * **description**: Validar que sea un título de 10 a 500 caracteres alfanuméricos. * **createDate**: formato de fecha. * **updateDate**: formato de fecha. * **softDelete:** formato de fecha o nulo.   **Atributo:**   * **id:** Un identificador único universal. * **name:** Validar que sea un título de 2 a 10. * **type:** Validar que solo reciba los siguientes valores Color, Talla, Marca, Fábrica. * **createDate**: formato de fecha. * **updateDate**: formato de fecha. * **softDelete:** formato de fecha o nulo. |
| **Restricciones** | 1. No se puede repetir un atributo en un mismo producto. 2. El código debe estar en inglés, documentado y con el estilo de escritura recomendado por la comunidad según el lenguaje o framework seleccionado.   **Nota**: No es obligatorio tener en cuenta todas las restricciones, pero cada una será un ítem evaluado. |
| **Importante** | 1. Una vez llegue la prueba a su correo empezará a contar el tiempo de ejecución de la misma. 2. Debe crear un repositorio público en **GITHUB,** en donde debe almacenar la prueba. 3. En el README, debe especificar con lenguaje técnico cómo debemos correr la prueba. 4. Una vez finalice la prueba técnica debe responder el mensaje de correo electrónico enviando con la misma, con la siguiente información:    1. Nombre completo.    2. ¿Cuánto tiempo gastó en hacer la prueba?.    3. ¿Terminó la prueba?.    4. ¿Qué dificultades tuvo?.    5. Link del repositorio público.    6. Video de nos más de 15 minutos en donde explique técnicamente el proyecto funcionando. |