

**Descripcion general**

2000s-Style es una tienda de venta de articulos personales con el concepto "moda rapida".

El menu general da la opcion de ingresar a una pagina de "quienes somos" con la ubicación en el mapa utilizando iFrame, un opcion para visualizar y comprar productos filtrando por categoría y accediendo a la descripcion ampliada al hacer click sobre la imagen del producto, junto con la posibilidad de ver las reseñas de los productos y un formulario de contacto. Desde la pagina de productos se puede interactuar con el carrito de compra, solo para usuarios registrados.

|  |  |
| --- | --- |
| Para comprar el usuario debe indentificarse logeandose en el sistema con su usuario y password. El acceso se valida contra la data persistente en mockapi. Si el usario logeado es administrador del site entonces se presenta un menu extendido con las opciones de Administracion donde se puede realizar CRUD sobre Productos y Usuarios.  Usar admin/password para test. |  |

|  |  |
| --- | --- |
| Toda la data que se se trata en el site recide en mockapi en el proyecto llamado SimulacionBackend  Todos los CRUD se realizan contra esta data persistida en mockapi |  |

**products** contiene los productos y su reseña en el campo reviews y el link a las imágenes en el campo thumbnail.

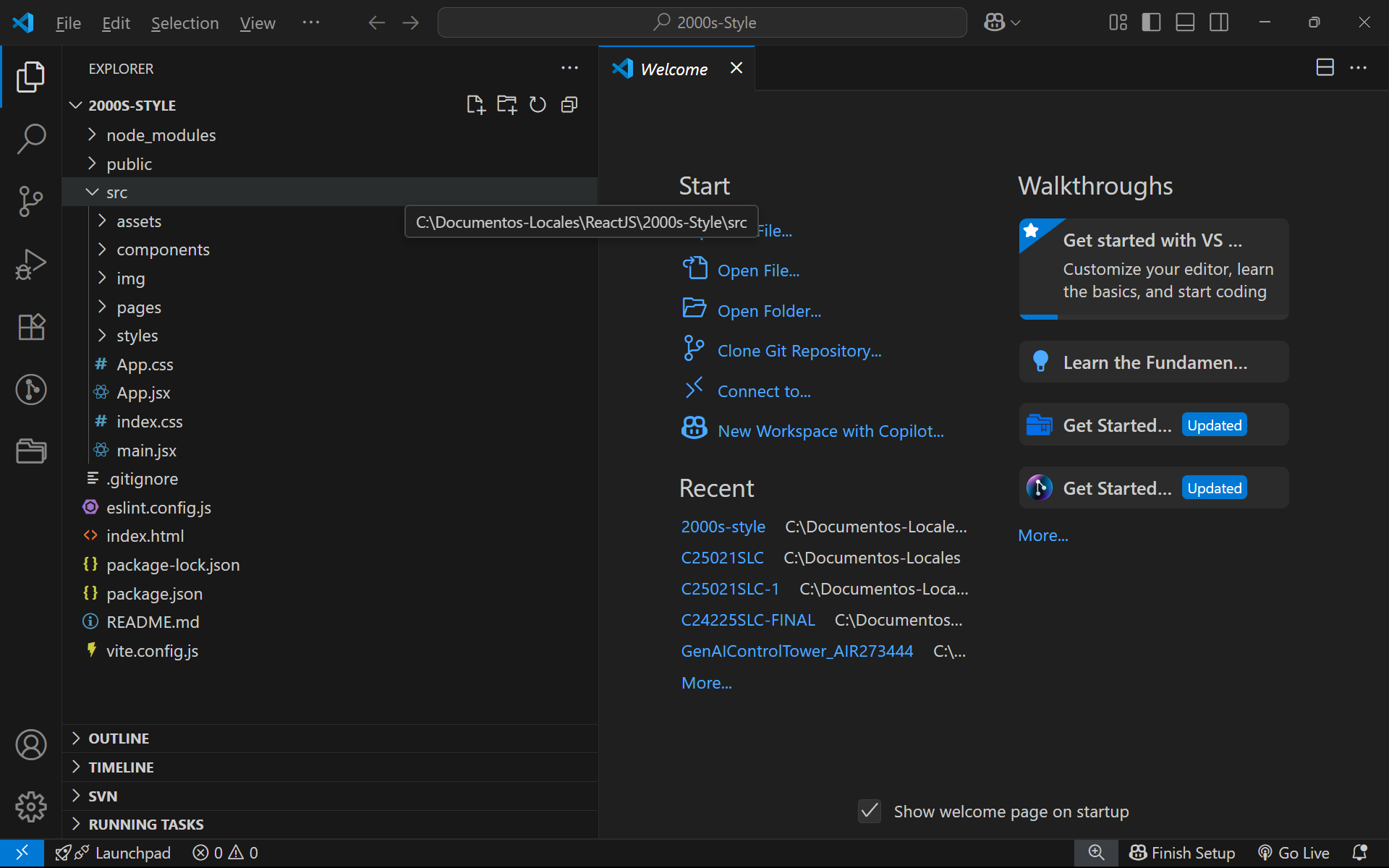
Esta data fue migrada de https://dummyjson.com/products.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| **users** contiene los datos de los usuarios del site. Entre esos datos el indicativo isAdmin para identificarlo como administrador.  Un usuario dado de alta como administrador es   * User: admin * Password: password |  |

**Estructura del proyecto**

|  |  |  |
| --- | --- | --- |
| El trabajo práctico se ha subido a un repositorio git | SI | <https://github.com/contesl/2000s-style.git> |
|  |  | Todo el codigo tiene comentarios para su mejor entendimiento |
| El proyecto se subio un servidor on-line para poder ser navegada por el Docente. | SI | <https://2000s-style.netlify.app/> |



|  |  |
| --- | --- |
| **Folder** | **Descripcion** |
| 2000S-STYLE | Raiz del proyecto. Contiene App.jsx y main.jsx |
| \components | Componentes reutilizables soporte en las diferentes paginas del site.  Incluye el componente para administrar contexto, filtros, footer y header de las paginas, el modal Bootstrap para el carrito de compras, la barra de navegación, las tarjetas para mostrar productos y reseñas, moda Bootstrap para mostrar la descripción detallada, el componente que administra la paginación en productos y reseñas y el componente para proteger rutas. |

|  |  |
| --- | --- |
| \pages  A screenshot of a computer  AI-generated content may be incorrect. | Se corresponden con cada punto del menu principal. Incluye las paginas de administracion de productos y usuarios, la pagina de contacto, de inicio, de login, de productos y reseñas como asi tambien la pagina “quienes somos”. |
| \styles  A screenshot of a computer  AI-generated content may be incorrect. | Style.css contiene todos los estilos del sitio. Se utiliza Bootstrap para estilos y responsividad |

**Consignas**

A screenshot of a computer

AI-generated content may be incorrect.

**Carrito de Compras con Context API.**

Toda la lógica del carrito de compras se administra dentro de \components\ModalCarrito.jsx, incluyendo el CRUD de itemes dentro del carrito. Se utiliza contexto

**Autenticación de Usuarios.**

|  |  |
| --- | --- |
| Solo los usuarios registrados pueden comprar. Si un usuario presiona el botón “Comprar” sin estar logeado se muestra el modal que se ve en pantalla. Esto ocurre en \pages\Productos.jsx.  Los usuarios pueden logearse a través del botón “Ingresar” del menú principal (\components\Navbar.jsx).  Los usuarios se validan contra la data existente en usuarios en mockapi.  Estan protegidas las rutas del carrito y de la administración, tanto productos como usuarios. |  |

A screenshot of a computer

AI-generated content may be incorrect.

**Formulario para Agregar Productos. Edición y Eliminación de Productos.**

|  |  |
| --- | --- |
| Si un usuario ingresó con su usuario y password y tiene perfil de administrador (isAdmin = true) en la data residente en mockapi entonces aparece el menú extendido con la opción Administración desde dónde se accede a un CRUD tanto de Productos como de Usuarios.  El formulario se valida y la persistencia se realiza en mockapi. |  |

**Manejo de errores.**

Se muestran mensajes de error en modales de Bootstrap o de validación de formularios en la misma interface según corresponda.

A screenshot of a computer

AI-generated content may be incorrect.

Todo el sitio utiliza Bootstrap para estilos y responsividad.

No se utiliza React Icons ni React Toastify.

No se realizo el trabajo de reconfigurar las etiquetas con React Helmet.

A screenshot of a computer

AI-generated content may be incorrect.

**Barra de búsqueda.**

Las paginas de Productos y Reseñas tienen filtro por categoría del producto.

Para los filtros se utiliza el componente reutilizable \components\FiltroCategorias.jsx

**Paginador de Productos.**

Tanto la página de Productos como la página de Reseñas utilizan paginación.

La paginación se administra con el componente reutilizable \components\ProductosConPaginacion.jsx.

A screenshot of a computer

AI-generated content may be incorrect.

**Pruebas de Compatibilidad.**

Se realizaron las pruebas de compatibilidad para todo tipo de dispositivos incluidos en Chrome y Edge en la opción de Developers Tool.

Adicional, ingresando a la aplicación desde los Smart pone de Apple, Samsung y Motorola.

**Optimización del Código.**

Todo el código está comentado y se ha tratado de “limpiar” todo duplicado a nivel código y a nivel archivo de estilos.

**Documentación Básica.**

Documento actual.