

| Evaluación de Proyecto Final |  |
|------------------------------|--|
| Estudiante:                  | Ulises Rodriguez   |
| Curso/Comisión:              | React JS - 47210   |
| Proyecto:                    | E-commerce App   |
| Consigna                     | Desarrollarás una app de un e-commerce para poder vender productos de un rubro a elección. |

| Nota final  |          |
|---|----------|
| 10  | Aprobado |
| ¡Felicitaciones! estás haciendo un gran trabajo.<br>¡Seguí así! |          |

| Criterios          | Descripción   | Alcance | Puntaje parcial | Devolución<br>Justificar porqué el estudiante se encuentra en ese nivel en cada uno de los criterios solicitados.  |
|--------------------|---|---------|-----------------|--|
| Convenciones       | Respetar todas las convenciones propuestas y lo hace en todas las instancias según lo propuesto.<br>Es consistente con sus declaraciones del readme.<br>Trata de afiliarse a las convenciones del ecosistema de React.<br>Presenta un linting consistente con espaciados que favorezcan la lectura                                    | Óptimo  | 2               | Excelente aplicación de todos los conceptos.   |
| Container patterns | Agrupar múltiples props de manera entendible y consistente bajo la consideración de respetar sus entidades.<br>Respetar namings a lo largo de la aplicación.<br>No agrega fetches extra dentro de los dummy components / componentes presentacionales.  | Óptimo  | 2               | Excelente aplicación de todos los conceptos.   |
| Navegabilidad      | Respetar las convenciones y los routings<br>Puede implementar patrones adicionales<br>Ejecuta navegaciones programáticas<br>Utiliza NavLink para ubicar al usuario y les da un styling que valide que ya está en una ruta<br>Si ya se encuentra en una ruta y navega a la misma ruta no re-ejecuta llamados                           | Óptimo  | 2               | Excelente aplicación de todos los conceptos.   |
| Catálogo y detalle | Filtrar por categorías implementando correctamente el useParams(:catId), y aprovecha un useEffect para detectar sus cambios de ruta y vuelve a disparar las tareas asíncronas.<br>Separa correctamente las responsabilidades.<br>Toda la información solicitada se muestra.<br>Usa loaders.<br>No sobre-optimiza y se mantiene simple | Óptimo  | 2               | Excelente aplicación de todos los conceptos.<br>Pequeña corrección, cuando necesites validar un array antes de un map, no uses array && array.<br>map((a) => etc.. Mejor validar por el .length > 0 .<br>También a la hora de nombrar una función o variable, intentar no abreviar por ejemplo "useGetFSC" |

|                                       |  |        |   |  |
|---------------------------------------|--|--------|---|--|
| <i>CartProvider /<br/>CartContext</i> | Mantiene las responsabilidades limpias<br>No lleva cosas de UI al context<br>No genera subestados para guardar totalizaciones o counts / summary<br>Puede crear otras variaciones de las sub-funcionalidades y adicionar funciones simples que mantienen la consistencia con el resto  | Óptimo | 2 | Excelente aplicación de los conceptos. |
| <i>CartWidget/Cart</i>                | No hay lógicas de cálculo de cantidades dentro del widget .<br>Puede separar el widget del rendering del icon.<br>Aparte de dividir responsabilidades es capaz de desarrollar un flow claro y reactivo de deshabilitación de campos.<br>Valida correctamente los inputs y sus tipos de datos.<br>No utiliza recursos como alert o nativos del browser manteniendo el | Óptimo | 2 | Excelente aplicación de los conceptos. |

