

Instrumento de evaluación de Proyecto Final	
Estudiante:	Completar con el nombre y apellido del estudiante evaluado
Curso/Comisión:	Completar con el nombre del curso y comisión
Proyecto:	Escribir el nombre del proyecto del estudiante
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. ¡Seguí así!	

Requisitos	Descripción	Alcance	Devolución	Comentarios del tutor
Autoría propia/originalidad	Tu trabajo debe ser personal y original y no presentar copia o plagio ni total o parcial de otros trabajos de Coderhouse o de la web. En el caso de haber plagio total, el trabajo no será corregido ni aprobado. En caso de haber una copia parcial en alguna de las partes del trabajo, se te pedirá rehacer dicha parte.	SI	OK	Completar
Entrega con permisos de accesos	El trabajo debe ser entregado con permisos de comentador para que el docente pueda acceder a su visualización. En caso de entregar un link privado o vacío, se debe solicitar al estudiante que entregue con permisos de accesos de comentador para que su trabajo pueda ser corregido	SI	OK	Completar
Formato solicitado	El trabajo es presentado bajo el formato solicitado (Google slides, Docs, Figma, Adobe XD, etc). En caso de no entregar en el formato solicitado, se debe pedir al estudiante la entrega bajo el formato establecido para la corrección de su trabajo.	SI	OK	Completar

Criterios	Descripción	Alcance	Puntaje parcial	Devolución 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. Es consistente con sus declaraciones del readme. Trata de afiliarse a las convenciones del ecosistema de React. Presenta un linting consistente con espaciados que favorezcan la lectura en	Óptimo	2	Completar
Container patterns	Agrupar múltiples props de manera entendible y consistente bajo la consideración de respetar sus entidades. Respetar namings a lo largo de la aplicación. No agrega fetches extra dentro de los dummy components / componentes presentacionales.	Óptimo	2	Completar

<i>Navegabilidad</i>	Respeta las convenciones y los routings Puede implementar patrones adicionales Ejecuta navegaciones programáticas Utiliza NavLinks para ubicar al usuario y les da un styling que valide que ya está en una ruta Si ya se encuentra en una ruta y navega a la misma ruta no re-ejecuta llamados	Óptimo	2	Completar
<i>Catálogo y detalle</i>	Filtra por categorías implementando correctamente el useParams(:catId), y aprovecha un useEffect para detectar sus cambios de ruta y vuelve a disparar las tareas asíncronas. Separa correctamente las responsabilidades. Toda la información solicitada se muestra. Usa loaders. No sobre-optimiza y se mantiene simple	Óptimo	2	Completar
<i>CartProvider / CartContext</i>	Mantiene las responsabilidades limpias No lleva cosas de UI al context No genera subestados para guardar totalizaciones o counts / summary Puede crear otras variaciones de las sub-funcionalidades y adicionar funciones simples que mantienen la consistencia con el resto	Óptimo	2	Completar
<i>CartWidget/Cart</i>	No hay lógicas de cálculo de cantidades dentro del widget . Puede separar el widget del rendering del icon. Aparte de dividir responsabilidades es capaz de desarrollar un flow claro y reactivo de deshabilitación de campos. Valida correctamente los inputs y sus tipos de datos. No utiliza recursos como alert o nativos del browser manteniendo el	Óptimo	2	Completar