| **Proyecto Frontend Angular (10)** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Criterios** | **Nulo**  AUSENTE  **0 PUNTOS** | **Bajo**  CONFUSO  **2 PUNTOS** | **En proceso**  FALTA DE PROFUNDIDAD  **3 PUNTOS** | **Correcto**  PROBLEMAS PUNTUALES  **4 PUNTO** | **Óptimo**  CLARO  **5 PUNTOS** | **NOTA** |
| Aspecto funcional | No se implementa un componente de login para acceder con cuenta de cliente o administrador.  El proyecto presentado no se adecúa a lo estipulado. | Se Implementa un componente de login para acceder con cuenta de cliente o administrador.  El menú lateral muestra las opciones correspondientes a cada perfil:   * admin: alumnos, cursos, inscripciones, usuarios, logout * usuario: alumnos, cursos, inscripciones, logout   Se trabaja con un backend en mockapi o similar pero su operación está incompleta. | Se Implementa un componente de login para acceder con cuenta de cliente o administrador.  El menú lateral muestra las opciones correspondientes a cada perfil:   * admin: alumnos, cursos, inscripciones, usuarios, logout * usuario: alumnos, cursos, inscripciones, logout   El toolbar superior muestra el nombre de la aplicación, el título del componente según la ruta y el nombre del usuario logueado  Las opciones de menú usuarios, cursos, y alumnos muestran los datos en una tabla cuando se seleccionan  Una vez logueado el usuario sólo se presenta la vista según su perfil.  El perfil cliente sólo lista las películas disponibles para su alquiler. No se presentan opciones en el carrito para borrar películas en forma individual o total.  Cada registro de la tabla tendrá una columna de acciones que el usuario podrá seleccionar sobre ese registro. Para todos serían la de modificar, eliminar, ver más detalle. Las opciones de modificar y eliminar sólo se deberán mostrar con un usuario con el rol de administrador | Se Implementa un componente de login para acceder con cuenta de cliente o administrador.  El menú lateral muestra las opciones correspondientes a cada perfil:   * admin: alumnos, cursos, inscripciones, usuarios, logout * usuario: alumnos, cursos, inscripciones, logout   El toolbar superior muestra el nombre de la aplicación, el título del componente según la ruta y el nombre del usuario logueado  Las opciones de menú usuarios, cursos, y alumnos muestran los datos en una tabla cuando se seleccionan  Cada registro de la tabla tendrá una columna de acciones que el usuario podrá seleccionar sobre ese registro. Para todos serían la de modificar, eliminar, ver más detalle. Las opciones de modificar y eliminar sólo se deberán mostrar con un usuario con el rol de administrador  La opción de ver detalle de un alumno se mostrará los datos del alumno más los cursos a los cuales está inscripto. Desde allí se podrá des inscribir de un curso a un alumno  La opción de logout deberá desologuear al usuario y llevarme a login | Se Implementa un componente de login para acceder con cuenta de cliente o administrador.  El menú lateral muestra las opciones correspondientes a cada perfil:   * admin: alumnos, cursos, inscripciones, usuarios, logout * usuario: alumnos, cursos, inscripciones, logout   El toolbar superior muestra el nombre de la aplicación, el título del componente según la ruta y el nombre del usuario logueado  Las opciones de menú usuarios, cursos, y alumnos muestran los datos en una tabla cuando se seleccionan  Cada registro de la tabla tendrá una columna de acciones que el usuario podrá seleccionar sobre ese registro. Para todos serían la de modificar, eliminar, ver más detalle. Las opciones de modificar y eliminar sólo se deberán mostrar con un usuario con el rol de administrador  La opción de ver detalle de un alumno se mostrará los datos del alumno más los cursos a los cuales está inscripto. Desde allí se podrá des inscribir de un curso a un alumno  La opción de ver detalle de un curso se mostrará los datos del curso más los alumnos a los cuales está inscripto. Desde allí se podrá des inscribir de un curso a un alumno  La opción de logout deberá desologuear al usuario y llevarme a login |  |
| Aspecto técnico | El proyecto no está publicado.  El proyecto no se vincula con lo entregado en instancias anteriores y no incorpora los elementos requeridos para esta instancia.  No hace ningún tipo de implementación del patrón Redux. | El proyecto presenta inconsistencias en el desarrollo de los elementos requeridos para las entregas anteriores e incorpora de manera incompleta los elementos requeridos para esta instancia.  Faltan componentes que hacen a la funcionalidad del proyecto.  No hay conexión entre el frontend y el backend.  Implementa un store a nivel de aplicación con NGRX para guardar el usuario logueado y el título a mostrar por cada componente en el toolbar  Utiliza acciones, selectores y reducers. | El proyecto presenta algunas inconsistencias en el desarrollo de los elementos requeridos para las entregas anteriores y contiene algunos errores en los elementos requeridos para esta instancia.  Se envía el código fuente del proyecto pero no existe una publicación.  El estudiante no notifica la imposibilidad de crear usuarios y/o no envía los datos de usuarios ficticios y/o del usuario y/o URL del perfil administrador.  El proyecto publicado en nube posee varios errores, que pueden rondar el 50% o más de las funcionalidades esperadas.  Implementa un store a nivel de aplicación con NGRX para guardar el usuario logueado y el título a mostrar por cada componente en el toolbar  Utiliza acciones, selectores y reducers. | El proyecto cuenta con los mismos elementos desarrollados en las anteriores entregas e incorpora lo detallado para esta instancia.  El proyecto se comporta mayormente de manera funcional, de acuerdo a los lineamientos solicitados para el proyecto final.  Implementa la librería NGRX y sus diferentes partes como faeture store, app store y effects  Algunos features modules tienen su propio feature store.  Algunos feature store implementan effects para utilizar las API de los servicios  La implementación de Effects no está correctamente implementada o no existe.  Implementa un store a nivel de aplicación para guardar el usuario logueado y el título a mostrar por cada componente en el toolbar  Utiliza acciones, selectores y reducers. | El proyecto cuenta con los mismos elementos desarrollados en las anteriores entregas e incorpora lo detallado para esta instancia.  El proyecto se comporta de manera funcional, de acuerdo a todos los lineamientos solicitados para el proyecto final  Contiene una excelente definición de la lógica utilizada y presenta una perfecta estructura.  Implementa la librería NGRX y sus diferentes partes como faeture store, app store y effects  Todos los features modules tienen su propio feature store.  Todos los feature store implementan effects para utilizar las API de los servicios  Implementa un store a nivel de aplicación para guardar el usuario logueado y el título a mostrar por cada componente en el toolbar  Utiliza acciones, selectores y reducers. |  |