

# RÚBRICA DE EVALUACIÓN DE ENTREGAS DEL PROYECTO FINAL

## Vue JS

Una **rúbrica** es un conjunto de criterios y normas que sirven para evaluar el nivel de desempeño en una tarea. Es una herramienta de calificación que simplifica la tarea de corrección y permite lograr una evaluación más objetiva y transparente tanto para el evaluador como el evaluado.

Adicionalmente, la rúbrica ayuda a igualar y alinear los criterios de evaluación entre los diferentes docentes y tutores.

### En resumen, la rúbrica:

1. Define los criterios de evaluación de cada entrega de proyecto final.
2. Describe y detalla los puntos clave que debe tener cada entrega.
3. Facilita y transparenta la corrección de cada trabajo.

**¿CÓMO SE UTILIZA LA RÚBRICA?**  [Ver el Tutorial](#)

## PRIMERA ENTREGA DEL PROYECTO FINAL

### Se debe entregar:

- Creación de un proyecto Vue CLI 2 con Bootstrap
- Creación de componentes: registro, login, listado, info y carrito
- Definir la lógica y estructura de representación de datos en listado, info y carrito utilizando tablas dinámicas tomando sus datos de archivos JSON locales ó de propiedades creadas dentro de los componentes con información de prueba.

**Consigna:** Crearás un proyecto frontend basado en Vue Js para hacer pedidos online de comida (o funcionalmente equivalente). Deberá contener el perfil de administrador, el cual podrá listar, realizar altas y bajas de los productos gastronómicos y modificar sus características. También podrá ver el pedido detallado de todos los clientes. El perfil cliente podrá ver el listado de todas las comidas y elegir una de ellas para enviarla a una vista de compra (carrito), y luego de establecer la cantidad que va a encargarse del producto elegido, el sistema calculará el precio a pagar por comida y el total, para la posterior validación y envío del pedido por parte del cliente.

Proyecto Vue CLI 2 (10)

Criterios	Nulo AUSENTE 0 PUNTOS	Bajo CONFUSO 1,5 PUNTOS	En proceso FALTA DE PROFUNDIDAD 3 PUNTOS	Correcto PROBLEMAS PUNTUALES 4 PUNTOS	Óptimo CLARO 5 PUNTOS	NOTA
Funcional	<p>No se presentó un proyecto Vue/Cli 2 o Vue creado con Vue CDN.</p> <p>No existe un framework CSS u hoja de estilos propietaria creada por el estudiante.</p> <p>La aplicación cuenta con parcial de vistas creadas:</p> <ul style="list-style-type: none"><li>listado de productos</li><li>carrito de compras</li></ul> <p>Ambas vistas cuentan con información vaga o parcial de productos listados y/o agregados al carrito.</p> <p>La aplicación no tiene botones de navegación entre vistas, ni tampoco incluye botones que permitan dar una idea de algún tipo de interactividad, como ser: <i>agregar un producto al carrito, incrementar cantidad de un producto, ver un detalle del producto, remover el producto del carrito.</i></p>	<p>El proyecto presentado fue creado con Vue/Cli 2 o con Vue CDN.</p> <p>El framework CSS u hoja de estilos propietaria tiene referenciado un framework u hoja de estilos CSS en el archivo index.html principal de la aplicación.</p> <p>La aplicación cuenta con un mínimo parcial de vistas creadas:</p> <ul style="list-style-type: none"><li>Listado de productos</li><li>Carrito de compras</li><li>Puede incluir información o detalle de producto</li><li>Puede incluir login / registro sin interacción alguna entre sí, o con el resto del proyecto</li></ul> <p>La <b>vista Listado de productos</b> (o equivalente) incluye productos representados en pantalla. Los mismos cuentan con una título y/o descripción básica. No tienen información coherente y/o puede faltar el botón Comprar/Agregar al carrito, y/o puede contar con el botón mencionado sin interactividad alguna.</p> <p>No existe una <b>vista información o detalle del producto</b>, o existe y no muestra información acorde que debería mostrar, quitando así un aporte de valor significativo al proyecto.</p> <p>No existen botones de regresar/volver/retornar en cada vista mencionada o existen sin interactividad alguna, teniendo el usuario de la aplicación que utilizar los botones de avanzar/retroceder del navegador web.</p> <p>La <b>vista Carrito de compras</b> (o equivalente) no cuenta con información</p>	<p>El proyecto presentado fue creado con Vue/Cli 2. Integra alguna alternativa de framework CSS o un archivo CSS creado por el estudiante, aplicado en algunos pocos elementos HTML que conforman las vistas de este proyecto.</p> <p>La aplicación cuenta con algunas de las vistas creadas:</p> <ul style="list-style-type: none"><li>Listado de productos</li><li>Información o detalle de producto</li><li>Carrito de compras</li></ul> <p>Las vistas mencionadas anteriormente cuentan con los elementos HTML mínimos necesarios para representar su funcionalidad.</p> <p>No existe una <b>vista login/registro</b>, o puede existir la vista login, sin una interactividad mínima necesaria, como ser un login mediante hardcoding o simplemente pulsando el botón asociado para trasladarse a la vista de Productos.</p> <p>La <b>vista Listado de productos</b> (o equivalente) incluye productos representados en pantalla. Los mismos cuentan con una título y/o descripción básica. Pueden incluir un botón Comprar/Agregar al carrito, el cual no tiene interactividad alguna.</p> <p>La <b>vista información, detalle del producto o equivalente</b> muestra información del producto similar a la que se ve en la vista Listado de Productos. No aporta valor significativo respecto a la vista principal.</p> <p>Pueden existir botones regresar/volver o retornar en cada vista mencionada, pero estos no tienen interactividad aplicada o existen los mismos teniendo el usuario</p>	<p>El proyecto presentado fue creado con Vue/Cli 2 e integra Bootstrap o alguna alternativa a este como framework CSS, el cual se ve aplicado, en casi todos los elementos HTML que conforman las vistas de este proyecto.</p> <p>La aplicación cuenta con la vistas creadas:</p> <ul style="list-style-type: none"><li>login</li><li>Listado de productos</li><li>Información o detalle de producto</li><li>Carrito de compras</li></ul> <p>Cada vista mencionada anteriormente cuenta con los elementos HTML necesarios para representar de forma clara su funcionalidad básica.</p> <p>La <b>vista login</b> cuenta con los campos usuario/password más un botón para identificarse.</p> <p>No incluye la <b>vista Registro</b> ni tampoco fue definida la misma como una sección de la aplicación.</p> <p>La <b>vista Listado de productos</b> (o equivalente) incluye productos representados en pantalla y estos cuentan con, al menos, un título, precio, y un botón Comprar o Agregar al Carrito.</p> <p>La <b>vista información o detalle del producto o equivalente</b> muestra el producto seleccionado con una descripción algo ampliada o similar a lo que se ve en pantalla. Puede contar con un botón volver o regresar para retornar a la vista de productos, principal.</p> <p>En estos últimos dos puntos, falta un detalle de cada producto y una imagen que lo ilustre. Puede faltar también un</p>	<p>El proyecto presentado fue creado con Vue/Cli 2 e integra Bootstrap como framework CSS el cual se ve correctamente aplicado, a través de sus diferentes elementos HTML, en las pantallas que componen dicho proyecto.</p> <p>La aplicación cuenta con las siguientes Vistas creadas:</p> <ul style="list-style-type: none"><li>login / registro (<i>por separado o combinadas en una única vista</i>)</li><li>Listado de productos</li><li>Información o detalle de producto</li><li>Carrito de compras</li></ul> <p>Cada vista mencionada anteriormente cuenta con los componentes o elementos HTML necesarios para entender su funcionalidad básica.</p> <p>La <b>vista Login</b> cuenta con la interacción mínima de campos (usuario/password), un botón para identificarse y un link a la Vista Registro, si es la primera vez en la cual el usuario ingresa.</p> <p>La <b>vista Registro</b> cuenta con la interacción mínima de campos (nombre, apellido, usuario (o email), password) más un botón para registrarse como tal.</p> <p>La <b>vista Listado de productos</b> (o equivalente) incluye una muestra de al menos dos o tres productos con una imagen, título, descripción, precio, y un botón denominado Comprar o Agregar al Carrito.</p> <p>La <b>vista información o detalle de producto o equivalente</b> muestra el producto seleccionado con una descripción mucho más amplia de su contenido, una o más fotografías que</p>	

		<p>del proceso que debe cumplimentar y/o tiene contenido generado de forma estática sin interactividad alguna.</p>	<p>de la aplicación que utilizar los botones de avanzar/retroceder del navegador web.</p> <p>La <b>vista Carrito de compras</b> (o equivalente) no cuenta con información del proceso asociado o lista al menos un producto con información básica o faltante de acuerdo a la funcionalidad que dicha vista debe tener. Puede contener botones de checkout sin funcionalidad asociada y/o muestra la información de manera estática, sin ningún tipo de interacción.</p>	<p>botón Agregar al carrito.</p> <p>La <b>vista Carrito de compras</b> (o equivalente) cuenta con información básica del proceso de los productos a comprar. Puede contener elementos agregados de forma estática, donde se lista al menos un producto con el importe también visualizado.</p> <p>Esta última vista puede no contener un botón de eliminar productos, o se incluyen pero sin funcionalidad.</p> <p>Puede contener un botón cerrar o volver o regresar a la vista anterior, sin interactividad.</p>	<p>describen al producto, el botón de comprar o agregar al carrito, y un botón volver o regresar, para retornar a la vista principal.</p> <p>La <b>vista Carrito de compras</b> (o equivalente) cuenta con información básica del producto que está comprando el usuario, cantidad, precio unitario y un cálculo del precio total en el caso que compre más de una unidad de un producto, o varios productos en la misma compra.</p> <p>Dispone de una opción para eliminar productos o reducir su cantidad.</p> <p>Cuenta con un botón cerrar o volver o regresar a la vista anterior.</p>	
Técnico	<p>No se presentó un proyecto.</p> <p>No cumple con las bases técnicas de Vue/Cli 2 ni de un framework CSS que aporte una estética coherente a las diferentes vistas del proyecto.</p> <p>No existen estilos CSS definidos en cada vista que permita una mínima definición estética de cada componente vue generado mediante templates.</p> <p>El nivel técnico general del código puede incluir algunas props y algún método. No incluye mixins, ni propiedades computadas, no existe reactividad en la aplicación, ni siquiera una mínima interacción generada mediante hardcoding.</p>	<p>El proyecto fue presentado a través de un link de Google Drive u otro medio no contemplado.</p> <p>El proyecto descargado incluye la carpeta <b>node_modules</b>, lo cual torna al mismo mucho más pesado respecto a cómo debería ser. Puede no contener la carpeta <b>node_modules</b> incluida pero, al tratar de agregar las dependencias y módulos mediante <b>node npm</b>, se encuentran errores en este proceso. Puede no haber sido creado con Vue/Cli 2 y solo incluir Vue CDN.</p> <p>No existe vista Login/Registro o existe con cero interactividad o código que permita algún tipo de interacción con el resto de la aplicación.</p> <p>La <b>vista Listado de productos</b> está generada de forma estática, no cuenta con un array JSON que provea los datos visualizadas y/o no existe posibilidad de seleccionar un producto para agregar al carrito de compras y/o no existe interacción entre el producto y una Vista detalle del producto. Faltan imágenes de cada producto y/o una descripción clara del mismo.</p> <p>La vista Carrito cuenta con datos estáticos y/o no permite agregar datos desde la vista Listado de productos.</p>	<p>El proyecto fue presentado a través de la plataforma Coderhouse o mediante un link a Github.</p> <p>El proyecto descargado incluye la carpeta <b>node_modules</b>, lo cual torna al mismo mucho más pesado respecto a cómo debería ser. Puede no contener la carpeta <b>node_modules</b> incluida pero, al tratar de agregar las dependencias y módulos mediante <b>node npm</b>, se encuentran errores en este proceso.</p> <p>Si existe la <b>vista login</b>, no hay interacción o código alguno que le de vida. Cuenta con un acceso directo a la vista principal de la aplicación.</p> <p>La <b>vista Listado de productos</b> (o equivalente) está generada de forma estática. No tiene un array JSON que provea los datos visualizados en pantalla. No existe la posibilidad de seleccionar más de una unidad del producto a comprar y/o no funciona el botón de agregar al carrito y/o no existe el acceso a una <b>vista Detalle de producto</b>.</p> <p>La <b>vista Carrito</b> (o equivalente) cuenta con datos estáticos y/o no permite agregar datos desde la vista Listado de productos.</p> <p>No contiene operaciones asociadas que</p>	<p>El proyecto fue presentado a través de la Plataforma Coderhouse o mediante un link a Github.</p> <p>El proyecto descargado no incluye la subcarpeta <b>node_modules</b>. Al abrir el proyecto con VS Code y ejecutar el comando de configuración inicial, se agregan los módulos y dependencias sin encontrar ningún error en dicho proceso.</p> <p>La interacción con la <b>vista Login</b> es directa. El usuario pulsa el botón asociado y la aplicación redirige al usuario a la vista principal. El estudiante puede haberle provisto al tutor, un <b>usuario y password estático hardcodeado internamente para que funcione el login de la aplicación</b> y esta conduzca al usuario hacia la Vista principal.</p> <p>La información ingresada en los campos login no se almacena internamente ni persiste dentro de otras vistas de la aplicación.</p> <p>La vista <b>Listado de productos</b> (o equivalente) tiene una interactividad mínima necesaria. Pueden cargar los datos a partir de un objeto JSON interno/externo, o haber sido generados estáticamente. La vista puede permitir agregar los productos a un array de datos o a una</p>	<p>El proyecto fue presentado a través de la Plataforma Coderhouse o mediante un link a Github.</p> <p>El proyecto descargado no incluye la subcarpeta <b>node_modules</b>. Al abrir el proyecto con VS Code y ejecutar el comando de configuración inicial, se agregan los módulos y dependencias sin encontrar ningún error en dicho proceso.</p> <p>Las <b>Vistas Login/Registro</b> interactúan correctamente entre sí de forma dinámica. Permiten ingresar datos, <b>validarlos</b> y/o alertar si algún dato ingresado falta o es incorrecto.</p> <p>La definición de los input types de login y registro es correcta, respetando el tipo de datos a ingresar por parte del usuario de la aplicación.</p> <p>La información ingresada en los campos de Login/Registro es correctamente almacenada en un array de datos JSON o props independientes dentro de la aplicación y hasta es persistente en otras vistas de la aplicación.</p> <p>La <b>Vista Listado de productos y Detalle de productos</b> es 100% funcional y dichos datos son cargados a partir de un objeto JSON interno de la aplicación o de un archivo JSON externo.</p>	

		<p>No existen operaciones asociadas que permitan recalcular importes, no existe posibilidad de eliminar productos ni de modificar su cantidad.</p> <p>El nivel técnico general del código puede incluir alguna props y algún método con interactividad mínima sobre la aplicación. No existen directivas, arrays JSON, propiedades computadas, mixins y/o casi todo el código interactivo incluido está realizado de forma hardcodeada, sin aportar valor alguno en la interactividad.</p> <p>Los templates creados utilizan algunas clases o propiedades CSS locales o pueden no utilizar nada de CSS que aplique estilos gráficos claros a la aplicación.</p> <p>Puede tener un framework CSS u hoja de estilos referenciada como CDN en el documento HTML index.</p>	<p>permitan recalcular el importe si es que existe la posibilidad de agregar productos desde la vista Listado de productos.</p> <p>No existe posibilidad de incrementar/decrementar la cantidad de un producto determinado, ni la opción de removerlo de la lista.</p> <p>El nivel técnico general del código incluye props, propiedades computadas y/o métodos. Puede contener alguna directiva mínima que permita generar una interacción a través de hardcoding o puede no incluir ninguna directiva.</p> <p>Los templates creados para las diferentes vistas utilizan clases o propiedades CSS definidas de manera local y/o no aprovechan las clases CSS propias del framework CSS u hoja de estilos CSS personalizada, que el estudiante incluyó en su proyecto. Puede tener un framework CSS referenciado hacia su CDN en el documento HTML index, y no incluido como dependencia e importado correctamente como tal.</p>	<p>prop y pueden ser los mismos que luego se muestran en la vista Carrito de compras.</p> <p>Los productos listados no permiten incrementar o decrementar su cantidad.</p> <p>La <b>vista Carrito</b> (o equivalente) cuenta con datos estáticos o dinámicos (partiendo estos últimos de un array JSON o props definida internamente en la aplicación).</p> <p>Puede contar con una interacción mínima, permitiendo eliminar elementos cargados y recalculando automáticamente el importe total del carrito.</p> <p>El nivel técnico general del código incluye props, arrays JSON, mixins, propiedades computadas y/o métodos, y algunas directivas para listar elementos en las diferentes vistas.</p> <p>Los templates creados para las diferentes vistas, cuentan con clases CSS propias de Bootstrap o del framework CSS alternativa, elegido por el estudiante. Dichas clases aportan una estética correcta a la aplicación Vue en general.</p> <p>Pueden incluirse clases o propiedades mínimas CSS creadas por el estudiante para modificar o complementar a las clases CSS estándar del framework CSS utilizado.</p>	<p>Los datos visualizados como productos conciden con lo que el array de datos almacena y lo que se muestra en pantalla.</p> <p>La vista de productos permite agregar los mismos a un array de datos JSON interno y hasta <b>incrementar/decrementar las unidades de cada producto agregado.</b></p> <p>La <b>vista Carrito</b> (o equivalente) cuenta con datos estáticos o dinámicos. La interacción de la vista carrito desde la vista Listado de productos es correcta. Puede contar con datos dinámicos donde el carrito se crea a partir de una interacción desde la vista Productos. También incluye operaciones mínimas necesarias para hacer el cálculo de costo del carrito de acuerdo a los productos agregados.</p> <p>Puede contener también la posibilidad de eliminar productos agregados y controla correctamente el cálculo de importe total o de la eliminación total de ítems.</p> <p>El nivel técnico general del código incluye props, arrays JSON. Genera interactividad general y utiliza para ella la Reactividad propia de la herramienta Vue.</p> <p>Se aplicaron Filtros de vista locales y/o globales, mixins, propiedades computadas, métodos, y condicionales y/o directivas para listar, mostrar, ocultar elementos en las diferentes vistas.</p> <p>Dentro de los templates creados para la aplicación en general, se aplicaron las clases CSS de Bootstrap, que permiten estilizar las diferentes vistas de la aplicación, haciendo que la misma luzca de manera profesional.</p> <p>Se incluyeron además, clases, propiedades u otros elementos CSS creados de manera individual, para personalizar los elementos base que Bootstrap ofrece.</p>	
--	--	---	--	---	--	--