

Proyecto Integrador

Entrega final

Fecha de entrega : 25/11

CONDICIONES GENERALES

El **proyecto integrador** se realiza en grupos **de 2 ó 3 integrantes**. Pueden elegir a sus compañeros de trabajo libremente. Deben confirmar con anterioridad los grupos de trabajo vía Slack hacia **la cátedra**. En caso de tener algún conflicto es fundamental que lo avisen previamente.

La fecha de la entrega final del proyecto integrador será el **25 de noviembre**. La entrega se realizará mediante repositorio en Github. Tendrán tiempo hasta las 23:59 del día de la entrega.

Se espera que los miembros de cada equipo distribuyan las tareas de forma equitativa, es decir, que todos deben hacer una cantidad similar abarcando todos los temas vistos. En caso de necesitar ayuda, por favor, avisen a sus profesores con la mayor anticipación posible.


Es importante que utilicen GIT durante el proyecto y no únicamente para la entrega. Estaremos observando el commit log y quienes realizan dichos commits. Esto **afecta tanto a la nota del trabajo como a la nota de concepto**. La ausencia de commits **durante el proceso** dará lugar a la desaprobación individual.


CRITERIOS DE APROBACIÓN

La **aprobación** de la entrega implica que el proyecto de manera global cumpla con los puntos de la consigna.

El proyecto **es completamente realizable** con los conocimientos y la sintaxis de código aprendida en clase. Esperamos verlo aplicado.

Cada punto a desarrollar tiene criterios propios de evaluación. Se analizará cómo lo desarrollaron para colocar el puntaje final.

 La utilización de código no realizado por ustedes (incluye herramientas de inteligencia artificial) implica la desaprobación inmediata del proyecto de manera grupal y la posibilidad de que la cátedra eleve el caso a la Universidad por motivo de plagio/copia.

 La **desaprobación del Proyecto Integrador** determina que todo el equipo accede a la instancia de recuperatorio. No accederán al recuperatorio y deberán recurrir quienes desaprobaban el parcial y esta entrega.

OBJETIVO DE LA ENTREGA

Desarrollar un catálogo de productos que tenga las funcionalidades para:

- Administrar productos.
- Administrar usuarios.

A diferencia de un ecommerce un catálogo de productos no cuenta con los datos de precios ni con la funcionalidad de carrito de compras. El proyecto deberá respetar el patrón MVC aprendido durante la cursada.

Para el desarrollo del proyecto se le dará a cada grupo un [proyecto base que pueden descargar aquí](#). Contará con la estructura HTML y el estilo del sitio en CSS. Cada grupo deberá utilizar esta base y completarla con las consignas del proyecto.

CONSIGNAS A DESARROLLAR

Orden y prolijidad

La organización de los archivos y carpetas debe ser de acuerdo a los recursos, tal como se desarrollo en clase, es decir, valoraremos en qué archivo se encuentra cada ruta y método controlador dependiendo del recurso al que responda.

El código dentro de cada archivo debe estar prolijo y legible, es decir, que esté correctamente indentado.

La nomenclatura adecuada de archivos y carpetas debe respetar lo visto en clase y que además respete nombres en minúsculas, sin espacios ni caracteres especiales.

La realización de commits debe mostrar evolución del proyecto a lo largo del tiempo.

Condiciona su aprobación el proyecto que presente el código en pocos commits sobre la fecha de entrega.

Migración a Node y Express

Deberán migrar todo el proyecto base a un nuevo proyecto utilizando Node y Express.

La estética del sitio debe modificarse de acuerdo a la temática del catálogo elegida por el equipo y mantener una línea en común a través de todas sus páginas.

Header y footer deben ser comunes en todas las páginas.

El footer deberá ser actualizado con la información correspondiente de cada grupo.

Tener en cuenta los siguientes puntos:

1. Crear las **rutas** necesarias para que todas las páginas del proyecto funcionen. Las rutas deberán utilizar controladores. Tener en cuenta que **los hipervínculos ya no redirigirán a un archivo .html**, sino a una **ruta** como las mencionadas en primer punto. Lo mismo aplica a los formularios.
2. Crear los **controladores** que se encarguen de la lógica de cada página. Los controladores deberán renderizar vistas.

3. Crear las vistas necesarias utilizando EJS.
4. Modificar las rutas de los **links** de CSS e imágenes de cada EJS para que funcione correctamente con la estructura que plantea Express (el uso de la carpeta **/public**).
5. Completar las siguientes páginas creando los formularios con al menos los campos indicados.
 - a. **Registración:**
 - i. email: campo de tipo email.
 - ii. usuario: campo de tipo texto.
 - iii. contraseña: campo de tipo contraseña.
 - b. **Login:**
 - i. email: campo de tipo email.
 - ii. contraseña: campo de tipo contraseña.
 - iii. recordarme: campo de tipo checkbox.
 - c. **Agregar producto:**
 - i. imagen: campo de tipo texto. Solo colocaremos el nombre del archivo. El archivo se guardará en la carpeta **/public/products**.
 - ii. Nombre del producto: campo de tipo texto.
 - iii. descripción: campo de tipo textarea.
6. El header y el footer deben estar armados de modo tal que al modificarlos se actualicen en todas las páginas.

Base de datos y modelos

Utilizando lenguaje SQL deberán:

- Crear la base de datos.
- Crear las tablas de la base de datos y sus claves foráneas.
- Insertar datos iniciales en las tablas:
 - 5 usuarios.
 - 5 productos: recordá que los productos deben estar relacionados con el usuario que los cargó. Puede haber varios productos de un mismo usuario.

El archivo SQL generado debe quedar guardado dentro de la carpeta principal del proyecto con el nombre **baseDeDatos.sql**. Permitirá a los profesores cargar en Workbench (o otro gestor MySQL) la estructura y acceder a los datos iniciales de la base que usaron en el proyecto.

Los campos de las tablas deben coincidir con la información pedida en los formularios correspondientes de login, registro y nuevo producto.

También deberán crear todos los archivos de modelos necesarios para el proyecto junto con las relaciones correspondientes. De mínima son 2 modelos: **Modelo User**, **Modelo Product**. Queda a criterio del equipo crear más modelos según lo requiera el proyecto.

Home page

La home del sitio debe mostrar los productos ordenados de forma descendente según su fecha de creación. El último producto agregado debe verse en primer lugar. Cada equipo definirá la cantidad de productos a mostrar.

Clickear sobre cada producto debe redirigir al detalle del mismo.

En cada producto debe figurar el usuario que lo cargó. El dato a mostrar será el **username**.

Registración

El catálogo debe permitir la registración de usuarios. El formulario pedirá al menos los siguientes datos y funcionalidades:

- **Email:** validar que es un campo obligatorio. Si el usuario envía el campo vacío debe recibir un mensaje especificando el error. No podrán registrarse emails duplicados.
- **Nombre de usuario:** campo obligatorio de tipo texto. Pueden repetirse los nombres de usuario.
- **Contraseña:** validar que es un campo obligatorio. Debe almacenarse en la base de datos de forma encriptada. Si el usuario envía el campo vacío debe recibir un mensaje especificando el error.

Los mensajes de error se mostrarán al usuario mediante ***res.send("El mensaje a mostrar")***. La página será accesible únicamente para los usuarios que **no estén logueados**. Queda a decisión del equipo a donde redirigirlo si quiere acceder.

Login

El catálogo debe contar con una página de login que debe cumplir con las siguientes características.

- El usuario debe poder ingresar utilizando su email y su contraseña.
 - **Email:** validar que el email exista en la db. De lo contrario el usuario debe recibir un mensaje especificando el error.
 - **Contraseña:** validar que la contraseña sea correcta. De lo contrario el usuario debe recibir un mensaje especificando el error.

Los mensajes de error se mostrarán al usuario mediante ***res.send("El mensaje a mostrar")***

- Una vez que el usuario se haya logueado el header en todas las páginas debe modificarse:
 - Saludando al usuario.
 - Los links de "Registración" y "Login" deben desaparecer del header.
 - Deben verse accesos a "Mi Perfil", "Logout" y "Agregar producto".

Buscá en el código de la etiqueta <header> y encontrarás la parte del código que debe mostrarse al estar logueado.

- La página de login será accesible únicamente para los usuarios que **no estén logueados**.

Logout

El catálogo debe permitir el logout completo del usuario destruyendo la sesión. Si el usuario quiere volver a loguearse tendrá que hacerlo desde la página de login.

Agregar un producto

Únicamente los usuarios logueados podrán **cargar un nuevo** producto. El formulario para cargar productos debe contemplar los campos definidos en la base de datos:

- 1) Imagen del producto: es un campo de texto con el nombre del archivo. Para que la imagen se vea en pantalla sugerimos construir en la vista la ruta hacia la carpeta que contenga las imágenes dentro de la carpeta **/public** y al final de la ruta se coloque el nombre del archivo proveniente de la base de datos.
- 2) Nombre del producto.
- 3) Texto descriptivo

Validar que estos 3 campos sean obligatorios. En caso de que no estén completos se debe avisar al usuario.

De forma automática debe almacenarse en la base de datos la fecha de creación del producto (created_at o createdAt).

Control de acceso

Implementar los controles de acceso para que:

- La página de login sea accesible únicamente si el usuario no está logueado. Si el usuario ya está logueado debe redirigir a
- El link y el formulario **"Agregar producto"** únicamente podrán verse si el usuario está logueado.

Punto 5 - Buscador de productos

El catálogo debe contar con un buscador de productos que deberá cumplir con las siguientes características:

- Buscar un producto indicando parte de su nombre.
- El resultado de búsqueda deberá mostrar los resultados en orden descendente (es decir, primero los posts más nuevos).

- En caso de no encontrar resultados debe mostrar el mensaje “No hay resultados para su criterio de búsqueda”.
- Al clicar en un producto debe mostrarse la página de **detalle** del producto.
- Los resultados de búsqueda deben mostrar el nombre del usuario que cargó el producto. Clicar sobre el nombre del usuario debe llevar al perfil del usuario.

NOTA: la cátedra se reserva el derecho de modificar o ajustar la consigna si fuera necesario.