

# Mercado Liebre

---

Seguramente recordaras Mercado Liebre 🤔. Es el proyecto que comenzamos a desarrollar al comienzo del curso, primero para darle estilos con CSS y luego para darle dinamica con EJS y datos provistos desde un JSON.

Ahora vamos a mejorarlo para implementarlo pero con base de datos.

## Planilla de planificacion

La planilla puede verla en este link

[https://docs.google.com/spreadsheets/d/1CrG3SiI7c5pNU2UbnMiiwm8iaUYsQV36PG1qqtb\\_oIM/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1CrG3SiI7c5pNU2UbnMiiwm8iaUYsQV36PG1qqtb_oIM/edit?usp=sharing)

Este documento sirve en primer lugar para que tanto ustedes como nosotros tengamos 100% claro los pasos a resolver y los criterios de evaluacion y aprobacion.

Tambien sirve para que ustedes, a modo de guia, como si hubieran hecho una planificacion en Trello.

Este documento tiene 5 columnas:

- Objetivo
- Tarea
- Puntaje
- Requerido
- Tiempo max. hs

Lo que indican estas columnas es:

- **Objetivo:** que es lo que queremos hacer con ese conjunto de subtareas
- **Evaluacion:** cada uno de los pasos para cumplir un objetivo y que vamos a evaluar
- **Puntaje:** cuanto vale el cumplimiento de cada una de esas tareas
- **Requerido:** si la tarea es obligatoria para aprobar ML o no (opcional).
- **Tiempo max. hs:** esta columna indica, solo a modo de referencia, cuanto seria el tiempo maximo en horas que deberia llevar resolver esa tarea especificamente. Esto quiere decir que si una tarea indica "2" y llevo 1h 40 min tratando de resolver un problema, tengo que pedir ayuda inmediatamente o me va a complicar el resto de las tareas! Hay tareas que alguien podria realizar en 15 minutos, pero nosotros estimamos como mucho 1 hora. Asi que, cuanto mas tiempo indique, tambien indica mayor complejidad. Medir el tiempo es esencial para no detener el avance y evitar el impacto sobre el resto.

# Proyecto base

Para comenzar te proveeremos un proyecto base que te permitira enfocarte solo en las tareas a realizar.

## Lo que te proveemos para empezar

- Un archivo sql con las tablas ya creadas y datos de prueba
  - El archivo de base de datos se llama `database.sql`
  - El archivo de datos se llama `data.sql`
  - No es necesario crear la base de datos, el script la crea
- Un proyecto inicializado con express generator
  - Tiene todas las carpetas y archivos de ejemplo
  - Las carpetas vacias son a modo de referencia por si necesitan usarlas
  - Pueden crear carpetas que necesiten libremente
- Todas las vistas de Mercado Liebre
  - Ninguna vista tiene EJS implementado al 100%
  - Reemplazar todos los bloques estaticos por dinamicos

## Lo que haras antes de comenzar

- Descargar el proyecto, descomprimir el zip e instalar dependencias
- Ejecutar el archivo SQL
- Configurar tu conexion local a MySQL

## Recordar

- Commits atomicos siempre
  - Mensajes descriptivos en los commits
  - Mantenerse dentro del tiempo maximo de referencia para preguntar sobre ese punto
- 

# Uso adecuado de Git y Github

Cuando hayas descargado el archivo que te proveemos con el proyecto inicializado, deberas seguir los siguientes pasos:

## Crear un repositorio en Github

Ingresa a tu cuenta de Github y crea un nuevo repositorio, y te sugerimos que utilices el nombre `mercado-liebre-entregable`.

En la opcion de visibilidad deberás elegir **Private**.

Luego **no debes seleccionar ninguna otra opcion** de Github, solamente crear el repositorio.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner \*

 alezvi ▾

Repository name \*

/ mercado-liebre-entregable ✓

Great repository names are short and memorable. Need inspiration? How about [curly-meme?](#)

Description (optional)

☐



**Public**

Anyone on the internet can see this repository. You choose who can commit.

☒



**Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐

**Add a README file**

This is where you can write a long description for your project. [Learn more](#).

☐

**Add .gitignore**

Choose which files not to track from a list of templates. [Learn more](#).

☐

**Choose a license**

A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

## Vincularlo con tu repositorio

Ahora deberas ingresar desde la terminal en la carpeta local donde descargaste el ejemplo base que te dimos y ejecutar en la terminal los siguientes comandos.

1. Vincular los archivos locales con el repositorio remoto

```
git remote add origin https://github.com/user/mercado-liebre-entregable.git
```

2. Ingresar a la rama main (**master ahora es main**)

```
git branch -M main
```

3. Enviar al repositorio el proyecto inicial

```
git push -u origin main
```

**Todo lo explicado hasta aquí es solo por esta única vez.**

## Para trabajar limpio

Para trabajar y lograr los resultados esperados se requiere enfocarse en objetivos específicos, no es posible resolver todo de una vez. Así como un curso no se hace en 1 día, un proyecto no se desarrolla de golpe, tiene etapas. En la columna A del excel que te compartimos están los objetivos principales y en la columna B los requisitos a cumplir.

### Crear una rama

Cada vez que comiences a resolver un objetivo del excel (columna A) y comiences a realizarlo, te sugerimos que hagas un branch nuevo.

```
git branch listado-de-productos
```

En esta rama solo vas a trabajar en cumplir todos los objetivos de este punto.

### Ingresar a la rama

Para comenzar a trabajar en esa rama debes ingresar primero;

```
git checkout listado-de-productos
```

### Salí de la rama

Evita la tentación de cambiar de rama para resolver otro punto de otro objetivo solo por que surgió algún impedimento para avanzar. Ese es el momento donde hay que buscar ayuda: en Playground, en Google, o en Discord al resto de los compañeros o a los profesores.

Salí de la rama a master solo cuando estes muy seguro de que cumpliste todos los objetivos definidos en el Excel.

Para salir de la rama de trabajo actual a main (antes master), en la terminal ejecutas el comando:

```
git checkout main
```

### Actualizá main (ex-master)

Una vez que ingresaste a main puedes traer los cambios de aquella rama y actualizar master.

```
git merge nombre-de-la-rama-anterior
```

De esta forma logramos que la rama principal (main) siempre tenga algo que funciona. No importa si esta incompleto, lo importante es que funcione. Y lo que esta en desarrollo se mantiene en ramas separadas que no van a ser fusionadas en main hasta no haber terminado de realizar todas las tareas de ese objetivo.

Si seguís estos pasos vas a tener menos dificultades en main o master. Vuelve a repetir estos pasos (crear, ingresar, salir, merge) todas las veces que comiences a resolver un objetivo (columna A).

## Commits atomicos

Un commit atomico es una actualizacion del repositorio que tiene un cambio o un grupo de cambios especificos.

Tomemos como ejemplo la primera tarea: "hacer un listado de productos".

Esa tarea puede descomponerse en subtareas:

- Hacer el HTML del listado
- Crear el modelo de Producto con todas las definiciones
- Hacer una ruta
- Crear el controller y usar el modelo con la vista
- Etc...

Entonces, tomamos una tarea por vez y cuando la terminamos hacemos los siguientes pasos:

1. Agregar cada uno de los archivos, explicando qué cambios se hicieron

```
git add views/index.ejs
```

2. Hacer el commit inmediatamente para explicar que cambios se hicieron en ese archivo

```
git commit -m "se agregó el listado de productos en el home"
```

3. Repetir los pasos anterior por cada archivo modificado
4. Pushear todos los cambios al repositorio remoto

```
git push origin main
```

El objetivo principal de estos pasos es aprender a planificar lo que vamos a hacer y acostumbrarnos a tomar una tarea por vez. Resolver 10 tareas de una vez es casi imposible. Se toma un punto específico, se determinan las acciones que se van a tomar para resolverlo, y se comienza a realizar el plan. Una vez cumplido, se hace público (add + commit + push).

---

## Requerimientos comunes

Los requerimientos generales que deben cumplir minimamente todos los ejercicios de Mercado Libre están en el Excel en el primer grupo de tareas **requeridas**.

En Mercado Libre buscamos que se pueda realizar todas las operaciones típicas de un CRUD.

- ☐ Listar todos los productos
- ☐ Crear productos
- ☐ Editar un producto
- ☐ Borrar un producto
- ☐ Registro
- ☐ Login

Todas las especificaciones de lo que vamos a evaluar de cada punto, están en el Excel (columna B).

---

## Requerido a elección

A continuación te presentaremos algunas ideas. No te pedimos que implementes todas, **solo deberas elegir una** e implementarla. Si quieres implementar otra más o todas, por supuesto que será mucho más valorado!

## Seguridad

Las rutas actualmente se encuentran desprotegidas, cualquier usuario puede acceder.

En este punto deberas implementar un middleware que permita validar que el usuario que ingresa a la página de perfil de usuario, es el mismo que está logueado. Es decir, que el usuario 1 no puede ver el perfil del usuario 2. **Importante:** no hay que enviar el id del usuario por URL ni por formulario.

## Perfil del usuario

Deberas crear una pagina a donde el usuario logueado pueda ver su informacion personal.

Actualmente la informacion que hay de usuario es muy poca. No importa la cantidad de informacion. Nos importa que puedas implementarlo, asi que no te preocupes por agregar datos a la base de datos ni hacer otras "magias" que terminen enredandote.

Con mostrar el email, el avatar y la fecha de registro que hay en la base de datos, suficiente!

## Paginado

Paginar quiere decir que en lugar de listar miles de productos de una vez, vamos a ofrecer lo mismo segmentado en paginas de 10 resultados.

La paginacion requiere que un parametro opcional de la url indique en donde comenzar el listado. Por ejemplo, cuando llegue por url a: `http://localhost:3000/products?start=20` El parametro `start` del query string indica cuantos resultados se debe "saltar". Es decir, si empezamos del resultado **1** al 10, o del **10** al 20, o del **20** al 30, etc... Solamente le indicamos desde donde empezar a listar, la cantidad siempre debe ser diez.

Tips:

- Deberas usar la funcion `findAndCountAll` de sequelize [Mas info aqui](#)
  - No es necesario aplicarlo en todos los listados, con uno es suficiente
  - En el front tiene que haber un link que diga "Ver mas resultados" para avanzar de página
- 

## Bonus

Te sobró tiempo ? Querés practicar mas ?

Te proponemos algunos puntos mas avanzados que podrías agregar para mejorar el resultado final.

Estos puntos **no son obligatorios**, pero si los haces obviamente estas sumando muchísimo.

Es posible que estos puntos tengan alguna complicacion adicional, ya que se debe requerir otras librerias y hacer configuraciones para que funcionen. Si ocurre **buscá ayuda** sin perder tiempo.

## Carousel de imagenes

En la pagina home donde estan los banners, o en alguna otra pagina donde haya multiples imágenes, podes agregar un carousel de imagenes que cambie la imagen cada X cantidad de tiempo, por ejemplo cada 3 segundos.

Te recordamos que esto se puede hacer solamente con los conocimientos que ya tenes de Javascript, utilizando las funciones temporizadoras de eventos como `setTimeout` o `setInterval`

Para mas informacion te recomendamos ver estos ejemplos [https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp)

Tambien podes usar alguna libreria de JS, te recomendamos <https://splidejs.com/image-slider/>

## Zoom de imagenes

En la pagina de detalle del producto podrias agregar un efecto de zoom. Es decir, que en la foto del producto el usuario pueda clickear y la misma se agrande para ver bien en detalle.

Te recomendamos que utilices la libreria Ligthbox. Mas informacion aquí <https://lokeshdhakar.com/projects/lightbox2/>

## Validacion frontend con validator js

En cualquier formulario, el que prefieras usar, aplicar validacion de todos los campos utilizando validator.js, la misma libreria que por detras utiliza express-validator.

<https://www.npmjs.com/package/validator>

---

## Evaluacion y aprobacion

La evaluacion del proyecto será despues del ultimo día del curso. Y el resultado se comunicará posteriormente por email junto con el resultado final del curso.

La aprobación de Mercado Liebre constituye una parte de los criterios de aprobación. Es decir, se suma a los requerimientos institucionales como mantenerse dentro del limite de faltas, y la participacion activa a lo largo de todos los sprints en el proyecto integrador.


En el Excel que acompaña a este documento estan detallados los objetivos, las tareas y el puntaje de cada tarea. Así que vos mismo podras medir si llegaste hasta el punto de aprobación. **Para aprobar se requiere al menos el 70% de las tareas requeridas resueltas.** Las opcionales no cuentan para el 70%. Pero recordá que la meta es llegar al 100%.

Junto con los objetivos requeridos hay un segundo grupo de tareas opcionales. Como dijimos, **deberas elegir una**, la que prefieras y realizarla.





De las tareas "bonus", es unicamente para quienes cumplieron el 100% de las tareas principales y ya realizaron la tarea opcional. No son obligatorias, son opcionales y son para quienes quieren un desafio mayor aun.


## Ayuda

Para solicitar ayuda puedes usar el canal oficial de la cursada y hablar con el resto de los compañeros y profesores. 

En los espacios de colearning que mantengamos podemos ayudar tambien.

Si preferis que te ayude Google o Stack Overflow controlá el tiempo, no te pierdas en búsquedas interminables olvidando a donde querias llegar realmente. Hablar con el grupo y los profesores es tu salvavidas. 

Tampoco te dejes llevar por cualquier solución que tal vez le haya servido a quien hizo esa pregunta, pero no sea tu caso. Busca soluciones y si no estas seguro, habla con los profes. 

Todo lo que solicitamos está dentro de lo explicado en Playground, así que te recomendamos recurrir allí. La documentación oficial de cada una de las herramientas que usamos tambien es un buen recurso, aunque a veces puede resultar difícil entenderla, pero en ese caso nuevamente tendras que consultar con los profes. 

Ahora si. Manos a la obra... 