Ejercicio 1: Lista de la compra para la cena del viernes

**Hacer una lista de la compra.**

Vamos a trabajar con dos repositorios remotos en este caso: **GitHub** y **GitLab**. Es decir haremos el mismo ejercicio utilizando estos dos repositorios. Vosotros decidiréis después de esto cual os gusta más, después de trabajar con ambos.

Objetivos

Practicar las funciones básicas de Git trabajando sobre un repositorio local y remoto.

Tareas

1. Crear Nuevo repositorio en el alojamiento elegido para el proyecto
2. Repositorio inicial con varios archivos:
   * **supermercado**.txt: lista compra supermercado
   * **carniceria**.txt: lista compra carnicería
   * **pescaderia**.txt: lista compra pescadería.

Inserta **varios productos** en cada una de las listas de la compra.

1. Configurar datos, para poder trabajar con VSCode en ese repositorio.
2. Crear una **rama adicional** "**cena\_viernes**". Crear una lista en esa rama (**asistentes\_cena\_viernes.txt**). Inserta algunos asistentes a la cena.
3. **Unir** (**mezclar**) los cambios de la rama cena\_viernes en master en el **repositorio local**.
4. Subir ambas ramas al **repositorio remoto** utilizado**.**
5. En el **repositorio local** , registrar **cambios adicionales** en las ramas master y cena\_viernes ( en ambas)en la lista asistentes\_cena\_viernes.txt
6. Incorporar (merge) los nuevos cambios en la rama cena\_viernes a la rama master. Resolver conflictos si aparecen. ¿Cómo lo has realizado?
7. Subirlo al repositorio remoto los cambios efectuados.

Ejercicio 2. Desarrollo con Git empleando un modelo centralizado con GitHub

**Organizar una barbacoa con Git empleando un modelo centralizado con varias ramas de desarrollo.**

Objetivo

En este proyecto debe crear un repo y añadir algún(os) colaborador(es) con el que elaborará una barbacoa. **Uno de los miembros** del equipo hace de ***coordinador***.

* Organizar una barbacoa con Git
* Modelo: centralizado, **todos tienen permisos** para modificar el repositorio.
* Repo inicial con varios archivos:
  + **comida.txt**: comidas a llevar
  + **bebida.txt**: bebidas a llevar
  + **materiales.txt**: materiales necesarios (sillas, mesas, etc.)
  + **asistentes.txt**: lista de personas que van a ir. Quien pueda llevar coche que lo indique y que se apunten abajo sus acompañantes.
  + **juegos.txt**: propuestas de juegos para llevar a la barbacoa.

Datos del repositorio

En el fichero README.md se tendrá la lista de recetas con los enlaces a los archivos de esta práctica.

Cada colaborador trabaja en una rama con su nombre en la que trabaja y cuando termina crea un [**pull request**](https://help.github.com/articles/about-pull-requests/) pidiéndole al **coordinador de proyecto** que **mezcle** los cambios en la rama en la rama master. El coordinador realiza la mezcla y actualiza el fichero README.md con un enlace a la(s) nuevo(s) archivos (s)

El trabajo se realizara por filas, siendo un integrante de la cada fila el coordinador, el resto serán colaboradores. Deberéis meter al profesor como colaborador en cada uno de vuestros proyectos.

El nombre de los repositorios se dará de la siguiente forma:

Barbacoa\_filaX, donde la X es el número de la fila 1,2,3…,por ejemplo Barbacoa\_fila1

Flujo de trabajo

* Clonar repo remoto (clone).
* La rama **master**  contiene la versión definitiva, sin errores.
* Los cambios **locales** se realizan sobre la rama de **desarrollo de cada uno de nosotros**.
* Resueltos los conflictos, si los hubiese, los cambios se integran en nuestra rama del servidor.
* Los cambios serán informados al **coordinador** mediante un **pull request**, el cual será el que proceda a hacer el merge en la rama master o no.

Tareas mínimas a realizar **por cada alumno**

* Clonar repositorio.
* **Modificar lista de asistentes** para **incluirse en la misma**.
* **Añadir** algún **dato** o hacer algún cambio en: bebidas, comidas y materiales.
* Repetir estos cambios dos o tres veces.
* Finalizar con una copia actualizada del repositorio.
* Deberán tener el rol de coordinador al menos una vez TODOS los miembros del equipo.

Forma de Evaluación

* Por cada grupo elegiré entre sus miembros, un coordinador, y simularemos (yo incluido) una situación de trabajo real, con los ficheros anteriormente mencionados.

Practica Final. Desarrollo de un proyecto en GitLab (**Puntuable**)

## Descripción

Desarrollar un proyecto colaborativo de **tema libre** en un alojamiento Git (**GitLab**).

Objetivo

* Aprender a colaborar con y mantener proyectos en servicios de alojamiento Git.
* **Modelo**: **centralizado** o **distribuido**. Colaboradores hacen repos derivados (*fork*) y contribuyen mediante solicitudes de integración de ramas (*merge requests*).

## Preparación

* Crear un grupo de 3 o 4 alumnos.
* Elegir un miembro del grupo como **gestor de integración**. Éste será el propietario del repositorio principal del proyecto y actuará como gestor y desarrollador. El resto de miembros del grupo serán sólo desarrolladores.
* **Elegir un tema para el proyecto** y plantear unas especificaciones y objetivos.
* **Integrar al profesor dentro del proyecto y asignarle el rol de master.**
* Crear el repositorio principal. Reflejar en el archivo README.md la descripción, especificaciones y objetivos del proyecto.

## Tema del proyecto

El tema del proyecto es libre. Debe elegirse un tema y objetivos alcanzables en una semana de trabajo (a tiempo parcial). El tema elegido será alguno de los contenidos que se estén estudiando en los distintos módulos que se imparten en el curso o algún contenido de módulos de cursos anteriores.

## Requisitos

* El proyecto debe basarse en el flujo típico de GitHub: rama master principal, desarrollo y correcciones en ramas temáticas con posterior fusión en master.
* Cada desarrollador debe hacer un número significativo de contribuciones (confirmaciones).
* Las confirmaciones deben ser correctas en cuanto a su contenido y descripción.
* Debe establecerse algún mecanismo de coordinación para el desarrollo del proyecto: e-mail, lista de corrreo, irc, etc.
* Todos los desarrolladores deberán usar en algún momento el sistema de **comunicación de errores (**bug report) y de discusión de solicitudes de fusión de la plataforma.
* El proyecto debe contar al menos con una **documentación mínima** que contenga la descripción del proyecto, los autores, la licencia y las reglas para contribuir al mismo.
* Es necesario que el resultado del proyecto **sea funcional**, aunque el proyecto no sea muy ambicioso.

Flujo de trabajo

El ejercicio se realiza siguiendo un flujo de trabajo con **varios desarrolladores** y un **gestor**.

* El **gestor** crea el repositorio principal en GitLab y crea algún contenido inicial.
* El/los alumnos abren una cuenta en GitLab.
* En GitLab crea un repositorio derivado del principal (*fork*) y lo hace público o interno.
* Clona su repositorio derivado localmente y **registra cambios** en una rama **distinta a "master**".
* Sube los cambios en la rama de desarrollo a su repositorio público en GitLab.
* Dentro de GitLab hace una solicitud de integración (*merge request* en terminología de GitLab) de la rama donde ha hecho sus modificaciones con la rama *master* del repositorio principal.
* Comprueba si su solicitud ha sido aceptada o rechazada,  lee los comentarios del gestor y actúa en consecuencia.
* El desarrollador actualiza su rama local master desde la rama master del repositorio principal para incorporar los cambios del gestor y de otros desarrolladores.
* Los desarrolladores pueden hacer tantas solicitudes de integración como quieran.
* Subir la **clave pública** **ssh** para facilitar el acceso al servidor.
* (Opcional) Instalar la aplicación **Labcoat** (Android) para acceder a GitLab desde el móvil.