

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
ANÁLISIS Y DISEÑO DE SISTEMAS 1  
SECCIÓN "A"  
ING. WILLIAM GUEVARA  
AUX: PEDRO CASTRO



# PRÁCTICA

GUATEMALA, 8 DE AGOSTO DE 2024  
SEGUNDO SEMESTRE

# ÍNDICE

<b>OBJETIVOS</b>	
OBJETIVO GENERAL .....	3
OBJETIVOS ESPECÍFICOS .....	3
<b>DESCRIPCIÓN</b>	
FEATURES .....	5
WORKFLOW .....	6
<b>CONSIDERACIONES</b> .....	7
ENTREGABLE .....	7
FECHA DE FINALIZACIÓN .....	7

## Objetivo General

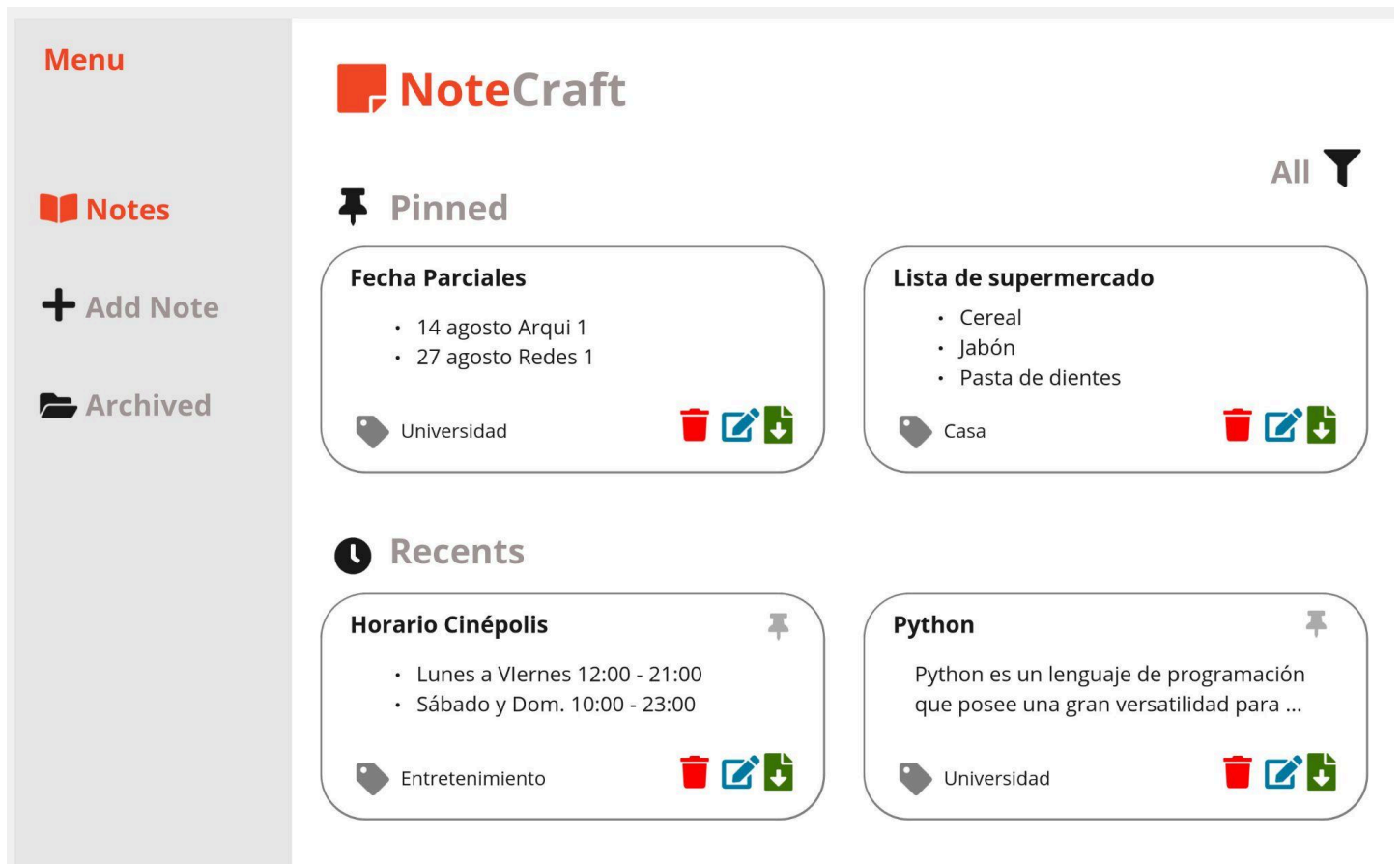
Desarrollar una aplicación web utilizando GitFlow como modelo de control de versiones, proporcionando a los estudiantes la experiencia práctica de trabajo en equipo y gestión de versiones en un proyecto de desarrollo.

## Objetivos Específicos

- Comprender los conceptos fundamentales de Git, incluyendo la estructura de un repositorio y cómo se modela el historial de cambios.
- Contribuir en un proyecto eficientemente aplicando buenas prácticas en el manejo de un sistema de control de versiones.
- Comprender la estructura de ramificación y flujos de trabajo asociados a GitFlow.

## Descripción

NoteCraft es un sistema que se desarrollará para la gestión de notas personales. El objetivo de esta App, es permitir al usuario el manejo de sus notas y apuntes, integrando funciones como agregar, modificar, eliminar, entre otras funcionalidades más (las cuáles se detallarán más adelante).



Propuesta de vista principal

## MANEJO DE BRANCHES

### Main/Master y Develop

Estas ramas se manejan de la siguiente manera:

- Main/Master: es la rama principal del repositorio, en esta rama se maneja un entorno de producción, es decir, solo se tendrá el código funcional.

- Develop: Esta rama se crea a partir de la rama Main/Master, de esta rama se crearán las ramas features. En esta se mantendrá el código antes de realizarse un release a la rama Main

## Release

Esta rama nace de la rama develop y se utiliza para pasar los cambios de una nueva versión a la rama main. Al fusionar la rama release con la rama main se solicita que se agregue una etiqueta a la rama main, con el nombre de la versión. (Al finalizar se espera que se tenga la versión 2.0.0). Asimismo, **la rama release también se debe fusionar con la rama develop.**

## Hotfix

Nace de la rama main para corregir un error crítico en producción, al finalizar la corrección, **esta rama se fusiona tanto en la rama develop como en la rama main**, y en la rama main se crea una etiqueta con el nuevo número de versión.

## Features

Para cada una de las funcionalidades se debe crear una rama con la siguiente nomenclatura "Feature/FeatureName\_#Id", donde FeatureName es el nombre de la funcionalidad a incorporar y Id el número de carnet. Al finalizar cada feature se debe fusionar a la rama develop.

A continuación se listan las funcionalidades esperadas para NoteCraft:

### Agregar Nota:

Tiene como objetivo almacenar una nueva nota en la lista, esta deberá almacenar los datos básicos que conforman la nota, las cuáles son:

- Título
- Descripción
- Etiqueta

Se deberá validar que para ingresar una nota, el título no deba ir vacío.

Una etiqueta se podrá crear en el mismo momento de agregar una nota nueva, o bien escogerla de las que ya fueron creadas en notas anteriores.

**Eliminar Nota:**

Permite remover una nota de la lista. Se recomienda que antes de realizar esta acción, el usuario pueda confirmar su decisión de remover la nota.

**Modificar Nota:**

Con esta funcionalidad el usuario puede seleccionar una nota y modificar cualquiera de los campos que fueron ingresados anteriormente, tomando en cuenta la validación de los campos mencionados en la feature de agregar.

**Fijar Nota:**

Esta opción permite darle una mayor prioridad a determinada nota, al seleccionarla, la nota aparecerá de primero en la página principal, en el apartado de “Fijado”.

**Archivar Nota:**

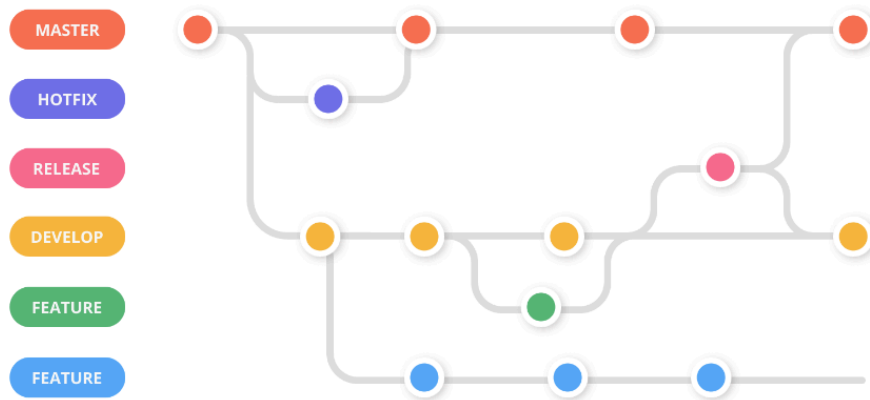
NoteCraft nos permite conservar las notas con menor relevancia sin tener que eliminarlas, por lo que se requiere tener una funcionalidad que permita archivar una nota, para que posteriormente esta nota pueda ser visualizada en la sección de Archivados. La nota archivada ya no debe aparecer en la página principal, únicamente en la sección de Archivados.

**Filtrar Notas:**

Esta funcionalidad permitirá filtrar las notas (por etiqueta) que se visualicen en la vista principal, este filtro aplicará tanto a las notas que se encuentren fijadas como a las que no.

## WORKFLOW

Se debe crear un Workflow de cómo quedarían las ramas con sus respectivos nombres y describiendo las acciones que se realizaron incluyendo el número de carnet del estudiante que lo realizó. Se recomienda utilizar herramientas como Miro o Draw.io; Este diagrama deberá estar incluido en el repositorio incluyendo la descripción de las acciones que se realizaron.



### CONSIDERACIONES:

- Es obligatorio el uso de una estrategia de branching (Gitflow).
- Se deben hacer al menos 2 releases a la rama Main.
- El lenguaje, frameworks, librerías para el desarrollo del backend y frontend queda a discreción del grupo.
- Se debe tomar en cuenta que la información de las notas se almacene en una base de datos, para ello, se deja a discreción del grupo si se utiliza de manera local o en la nube.
- Aplicar buenas prácticas para el uso de los commits.
- Deben agregar al auxiliar con rol de colaborador en su repositorio.

GITHUB	GITLAB
PedroCastro2001	PedroCastro2001

- Las copias totales y parciales tendrán una nota de 0 y será reportada a la escuela de sistemas.
- Todos los integrantes deben estar presentes el día de la calificación, ya que mediante actividades se validará la participación de los integrantes en la práctica.

### ENTREGABLES

- Se debe adjuntar el link del repositorio con el nombre [AYD1]Practica\_G#, entregar vía UEDI.
- El repositorio debe contener tanto el código fuente como el diagrama del flujo de las ramas.

**FECHA DE FINALIZACIÓN:** 22 DE AGOSTO DE 2024 A LAS 23:59