



**UNIVERSIDAD
DE LA FRONTERA**

E.T.I.C.O

Joaquin Pizarro, Fernando Torres, Gonzalo Fuentes

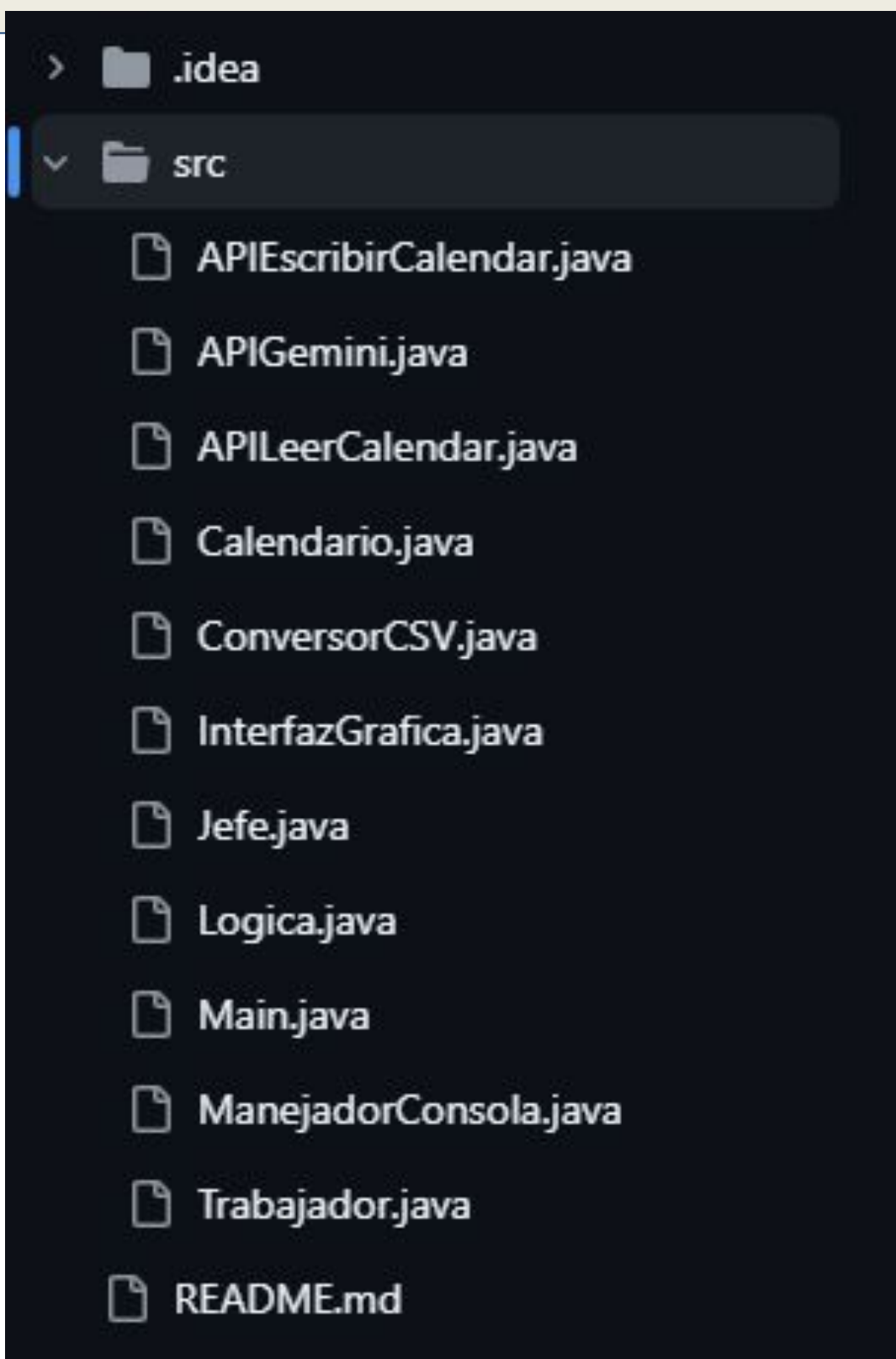
Contexto

La recurrente dificultad que presentan los grupos de trabajo a la hora de encontrar un periodo al avanzar proyectos o simplemente trabajar causa muchos problemas.

Los cuales terminan derivando en avances deficientes y mala coordinación entre un mismo equipo de trabajo.

Debido a esto hemos tenido la idea de crear un programa el cual se encargará de con los horarios de cada integrante del grupo encontrar un horario adecuado para instaurar un horario de reunión/trabajo o avances.

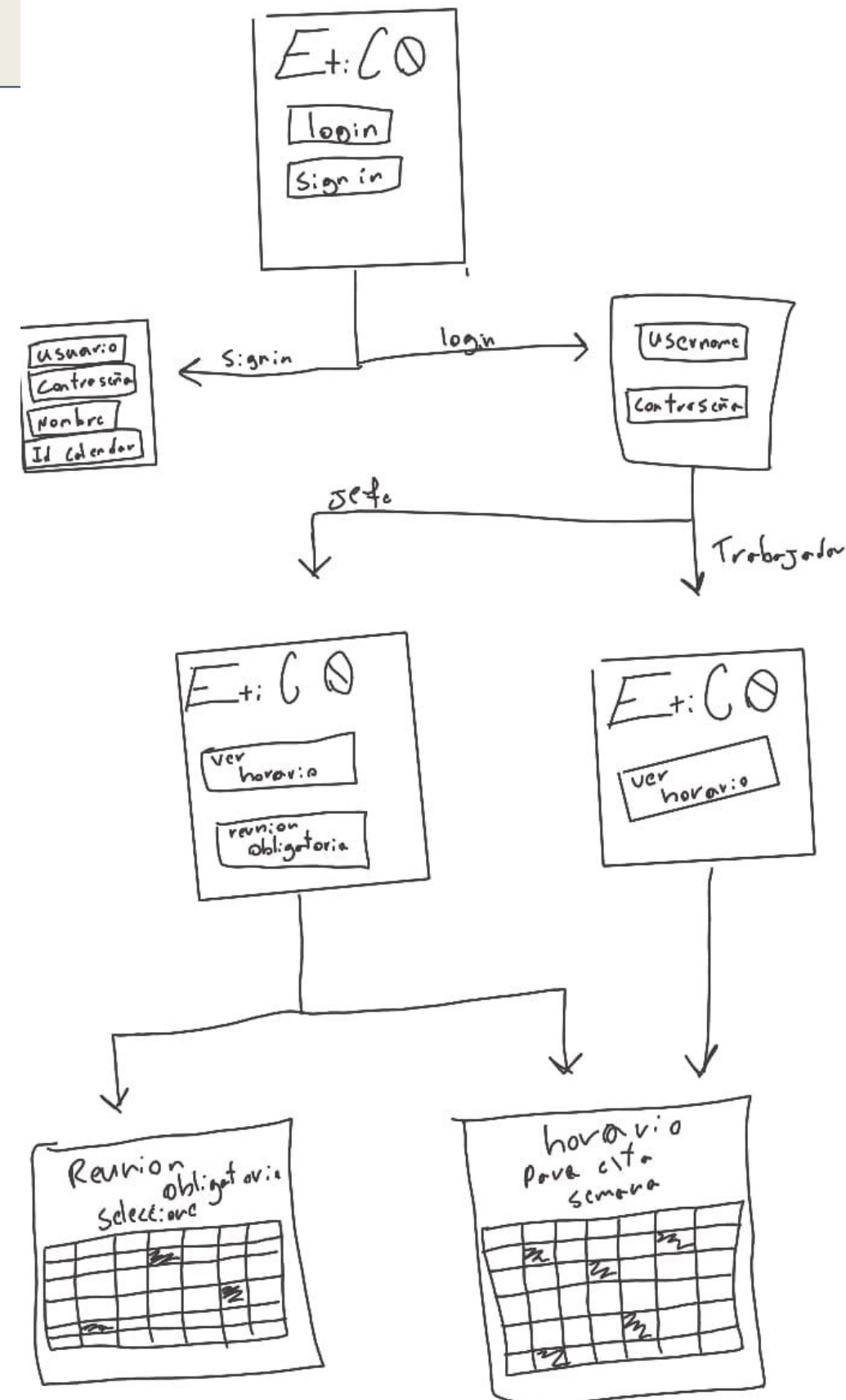
Arquitectura actual del proyecto



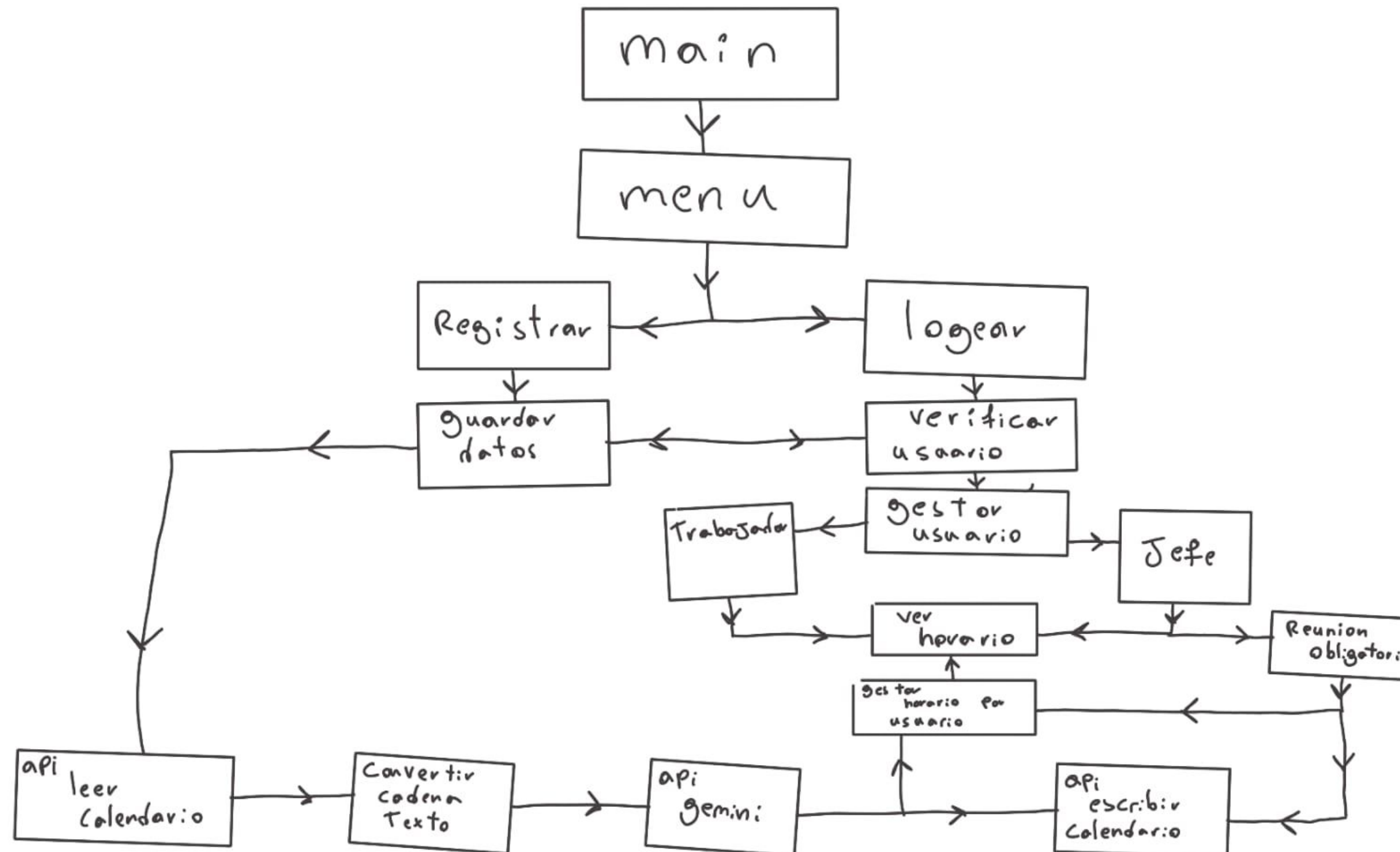


UNIVERSIDAD
DE LA FRONTERA

Idea de GUI



Flujo esperado del proyecto



Prueba de programa

```
--- Iniciando el proceso para agendar la reunion: 'Reunion de Sincronizacion del Proyecto' ---  
Simulación: Obteniendo horarios ocupados para jefe@ejemplo.com  
Simulación: Obteniendo horarios ocupados para trabajador1@ejemplo.com  
Simulación: Obteniendo horarios ocupados para trabajador2@ejemplo.com  
--> Horarios ocupados de todos los participantes recopilados  
Simulación: Enviando prompt a Gemini para encontrar el mejor horario.  
--> La IA ha sugerido el siguiente horario: 2025-10-25T11:00:00-11:30:00  
Simulación: Creando evento en el calendario de jefe@ejemplo.com  
Detalles: Reunion de Sincronizacion del Proyecto a las 2025-10-25T11:00:00-11:30:00  
Simulación: Creando evento en el calendario de trabajador1@ejemplo.com  
Detalles: Reunion de Sincronizacion del Proyecto a las 2025-10-25T11:00:00-11:30:00  
Simulación: Creando evento en el calendario de trabajador2@ejemplo.com  
Detalles: Reunion de Sincronizacion del Proyecto a las 2025-10-25T11:00:00-11:30:00  
  
--- Reunion agendada con exito para todos los participantes. ---  
  
Process finished with exit code 0
```

Guia de usuario

Paso 1: Ingresar datos de los integrantes, Las fechas se extraen de "APILeerCalendar.java"

Paso 2: Los datos se almacenarán para el programa y se envían a la "APIGemini.java"

Paso 3: El programa mostrará la mejor coincidencia de horario en la consola y hace envío de un evento a los calendarios de los integrantes.

Conclusión

Aún nos queda dividir el proyecto en packages para la buena organización de clases, encontrar una alternativa distinta para la API de IA, avanzar la GUI y terminar de estructurar el resto de clases, con respecto a la API del programa.

Pese a estos contratiempos, se tiene bien estructurado el modelo a seguir del programa y solo restan dificultades a la hora de realizar el código, pues el funcionamiento de este ya está planteado.