

PARADIGMAS DE PROGRAMACIÓN

PROYECTO SEMESTRAL DE LABORATORIO

Versión 1.1 - actualizada al 22/09/2024. Se agrega instrucción de como usar Google AI Studio (Google Gemini)

Versión 1.0 - actualizada al 17/09/2024

Versión Preliminar - actualizada al 2/09/2024

1.Descripción general

Durante el semestre se trabajará en distintos paradigmas de programación, los cuales serán puestos en práctica mediante distintos lenguajes de programación en tres entregas de laboratorio. El tema de laboratorio de este semestre se enfoca en la implementación del juego Conecta 4, también conocido como Cuatro en Raya (en inglés, Connect4).

Los estudiantes desarrollarán soluciones a partir de los conceptos y principios de distintos paradigmas de programación. Estas soluciones permitirán a un usuario jugar Conecta 4 y realizar distintas operaciones relacionadas con el juego. El software en su versión final (Laboratorio 3) tendrá una interfaz de usuario a través de la cual se podrá jugar Conecta 4 a través de la línea de comandos.

Los aspectos de implementación del laboratorio quedan abiertos a la creatividad del estudiante, siempre cumpliendo con los requisitos obligatorios, los cuáles se presentan en las siguientes secciones y cada enunciado de laboratorio tiene un conjunto de requerimientos que debe ser abordado bajo el paradigma correspondiente.

Enunciado¹

Conecta 4 es un juego de estrategia para dos jugadores que se juega en un tablero vertical de 6 filas y 7 columnas. Cada jugador tiene a su disposición 21 fichas, cada jugador tiene asociado un color diferente. Los jugadores se turnan para dejar caer fichas de su color (generalmente **rojo** y **amarillo**) desde la parte superior del tablero. Las fichas caen hasta la posición más baja disponible en la columna seleccionada. El objetivo del juego es ser el primero en formar una línea de cuatro fichas del mismo color, ya sea horizontal, vertical o diagonalmente.

El sistema que usted implementará debe permitir a los usuarios realizar operaciones como las siguientes:

1. Creación y Gestión de Partidas:
 - Iniciar una nueva partida.
 - Guardar el estado de una partida en curso.
 - Cargar una partida guardada.
 - Configurar el modo de juego (jugador vs jugador, jugador vs IA).
2. Gestión de Jugadores:
 - Registrar nuevos jugadores con nombre y color preferido.
 - Modificar la información de los jugadores existentes.
 - Consultar estadísticas de los jugadores (partidas ganadas, perdidas, empatadas).
3. Desarrollo del Juego:
 - Realizar movimientos (colocar fichas en el tablero).
 - Validar movimientos según las reglas del juego.
 - Detectar victoria, empate o continuación del juego.
 - Mostrar el estado actual del tablero.
4. Implementar integración con un proveedor de Gen AI para las siguientes funcionalidades:
 - Modo de juego usuario vs máquina IA, donde la IA corresponde a la respuesta de una consulta desde el programa hacia la API gratuita de Google AI Studio (Google Gemini).
 - Estadísticas del juego, en lenguaje humano haga un resumen de las estadísticas del juego contemplando las victorias y derrotas de cada jugador
5. Análisis del Juego:
 - Generar un historial de movimientos de la partida.
 - Permitir retroceder movimientos (función "undo").
6. Sistema de Puntuación y Ranking:
 - Implementar un sistema de puntuación basado en victorias.
 - Mantener un ranking global de jugadores basado en cuantas victorias han obtenido.

Lo anterior es sólo el contexto del problema, para detalles de implementación y lo pedido en cada entrega debe referirse al enunciado particular de cada laboratorio.

¹ Solo esta parte de la narrativa del enunciado fue elaborado con la asistencia de Chat GPT 3.5 para alguna de sus partes.

A modo de ejemplo, la animación de la Figura 2 muestra el funcionamiento del juego con dos jugadores, amarillo y rojo, donde este último después de múltiples jugadas alcanza la victoria a través de conectar cuatro fichas diagonalmente.

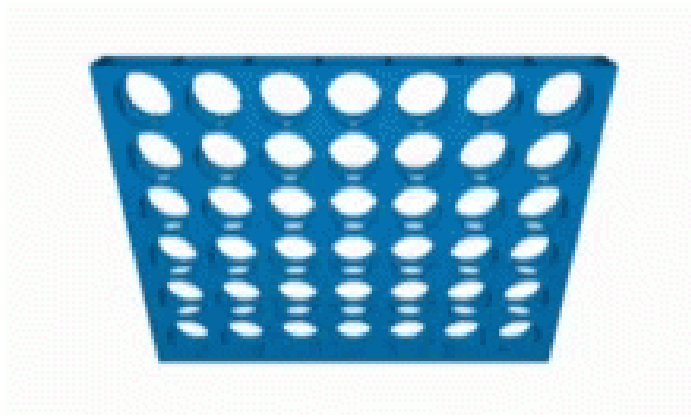


Figura 2: Ejemplo de juego en Conecta4

Fuente: Imagen obtenida desde [Connect Four - Wikipedia](#) (Fecha de consulta Lunes 2 de septiembre)

Instrucciones del juego Conecta4

El juego comienza con un tablero vacío de 7x6 con dos jugadores como mínimo. Cada jugador tiene a su disposición 21 fichas, número que disminuye para cada jugador al momento de realizar una jugada, lo cuál se ejemplifica en la Figura 3 y 4.

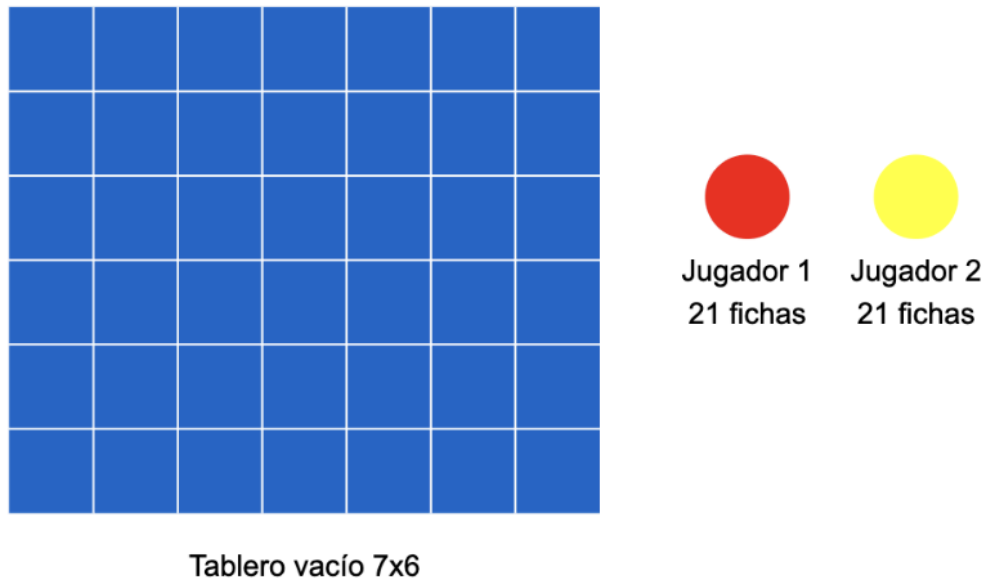


Figura 3. Inicio del juego

Fuente: Diagrama generado mediante claude.ai

El juego contempla dos jugadores (jugador 1 y jugador 2). Cada jugador selecciona un color de ficha (generalmente rojo o amarillo). Luego de la selección cada jugador comienza con 21 fichas asociadas a cada color, donde la cantidad de fichas asociada a cada jugador disminuye por cada jugada en el turno del jugador asociado (Hasbro, s.f.).

Quien inicia el juego no es relevante para el contexto y resolución de este laboratorio. Un ejemplo de jugadas se evidencia en la Figura 4.

Cómo Jugar Conecta 4

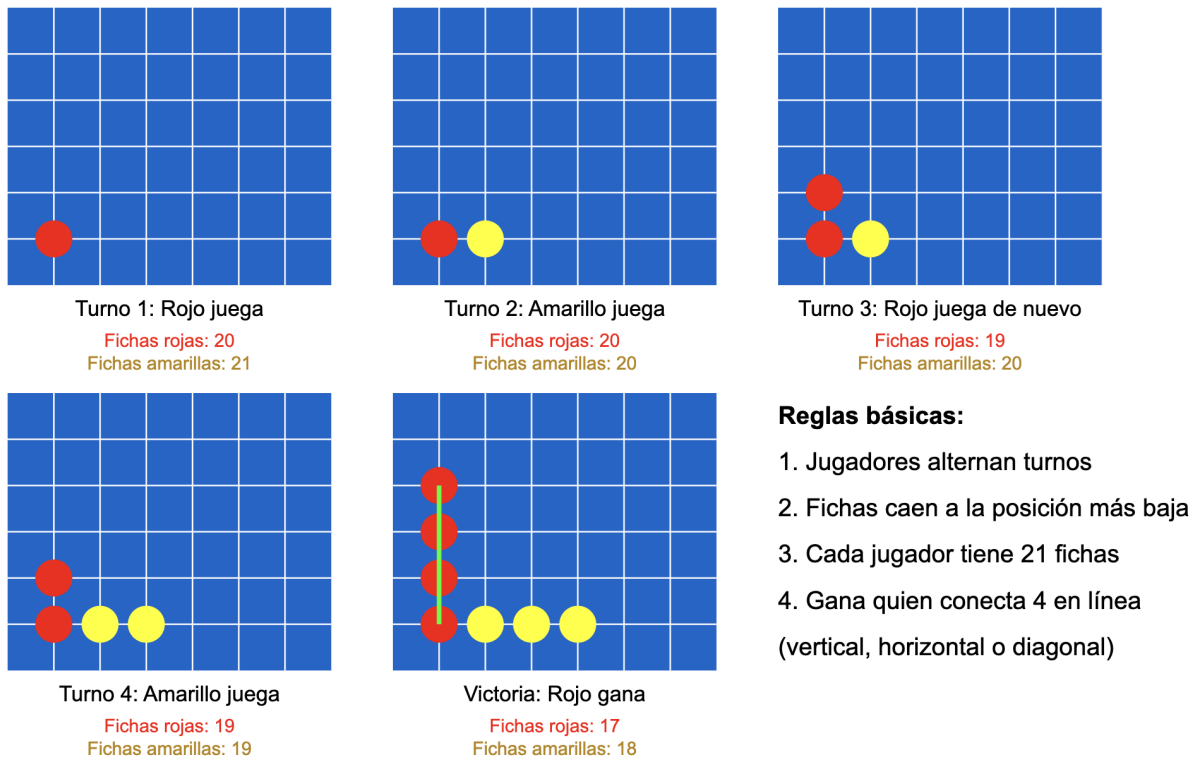


Figura 4. Ejemplo de juego Conecta4
Fuente: Diagrama generado mediante claude.ai

En la figura anterior se evidencia las siguientes reglas de juego:

1. Lineamientos generales del juego:
 - El juego comienza con un tablero de 7 columnas y 6 filas (7x6) el cuál se inicia vacío.
 - Cada jugador tiene asociado un color. Luego cada jugador tiene a su disposición 21 fichas del color asignado.
 - En cada turno los jugadores deben realizar alguna jugada con algunas de sus fichas.
 - En caso de que el tablero no permita realizar jugadas o si algún jugador se queda sin fichas entonces el juego se declara en empate.
2. Juego:
 - Turno 1: **El jugador rojo** coloca su primera ficha. En este caso particular selecciona jugar en la primera columna, por lo que la ficha cae hasta el fondo de la primera columna (**última posición disponible de la columna**).
 - i. **Fichas rojas restantes: 20**
 - ii. **Fichas amarillas restantes: 21**
 - Turno 2: **El jugador amarillo** responde jugando su ficha correspondiente en la siguiente columna, cayendo hasta la última posición disponible de dicha columna.
 - i. **Fichas rojas restantes: 20**
 - ii. **Fichas amarillas restantes: 20**

- Turno 3: Rojo juega de nuevo en la misma columna que jugó anteriormente. La ficha roja cae hasta la última posición disponible de dicha columna, que en este caso es la penúltima de la primera columna.
 - i. **Fichas rojas restantes: 19**
 - ii. **Fichas amarillas restantes: 20**
- Turno 4: Amarillo juega en la siguiente columna respecto a su última jugada. La ficha cae hasta la última posición disponible de dicha columna.
 - i. **Fichas rojas restantes: 19**
 - ii. **Fichas amarillas restantes: 19**
- **Turno N:** Luego de varias jugadas, Rojo en su jugada del turno actual juega su ficha en la primera columna que cae en la primera posición disponible. **Luego Rojo gana al conectar cuatro fichas verticalmente.**
 - i. **Fichas rojas restantes: 17**
 - ii. **Fichas amarillas restantes: 18**

3. Consideraciones:

- **Si ningún jugador puede realizar un movimiento dentro del tablero, es decir, una nueva jugada o conectar cuatro fichas y las fichas se han acabado, el juego se declara como empate.**

Formas de ganar en Conecta4

Existen tres formas en las cuales un jugador puede ganar el juego de Conecta4. La victoria se da cuando un jugador logra reunir 4 fichas de forma diagonal, vertical o horizontal, tal como se ejemplifica en la Figura 5.

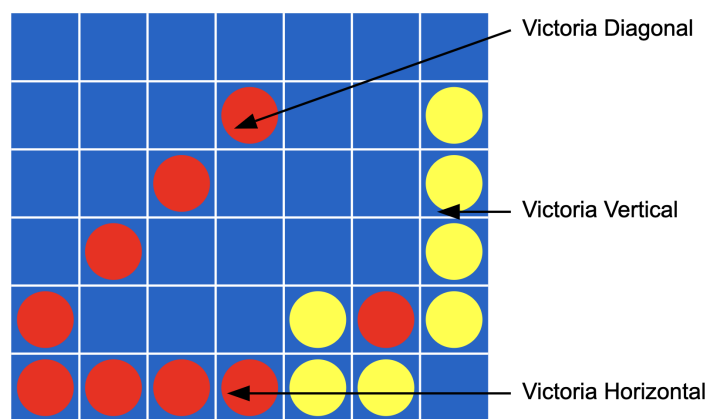


Figura 5. Formas de ganar Conecta4

Fuente: Diagrama generado mediante [claude.ai](https://www.claude.ai)

Referencias

Hasbro. (s.f.). Connect Four [Instrucciones del juego].
<https://www.hasbro.com/common/instruct/ConnectFour.PDF>

Integración con Google Gemini API

Cada enunciado de laboratorio en particular, ya sea funcional, lógico o orientado a objetos, un requerimiento indica que se debe realizar una integración con un servicio de inteligencia artificial generativa (GenAI), en este contexto será Google Gemini. La razón es que es una de las plataformas que permite uso gratuito sin ingresar tarjetas de crédito ni información personal. Actualmente no se puede realizar integraciones con ChatGPT debido a que este último pasó a ser de pago incluyendo su capa más básica.

Para lograr una integración Google Gemini primero debe ingresar a [prompt | Google AI Studio](#) usando su cuenta de correo Usach o su correo electrónico personal gmail. Para este laboratorio no es relevante con que tipo de correo gmail crea una cuenta en Google AI Studio.

Posteriormente se mostrará en pantalla el inicio de Google AI Studio (Figura 6). Esta es una plataforma con la cuál usted puede obtener algo llamada: API KEY. Una API KEY es un clave propia generada por la plataforma para su cuenta y permite a distintos programas, incluyendo su laboratorio interactuar con la plataforma.

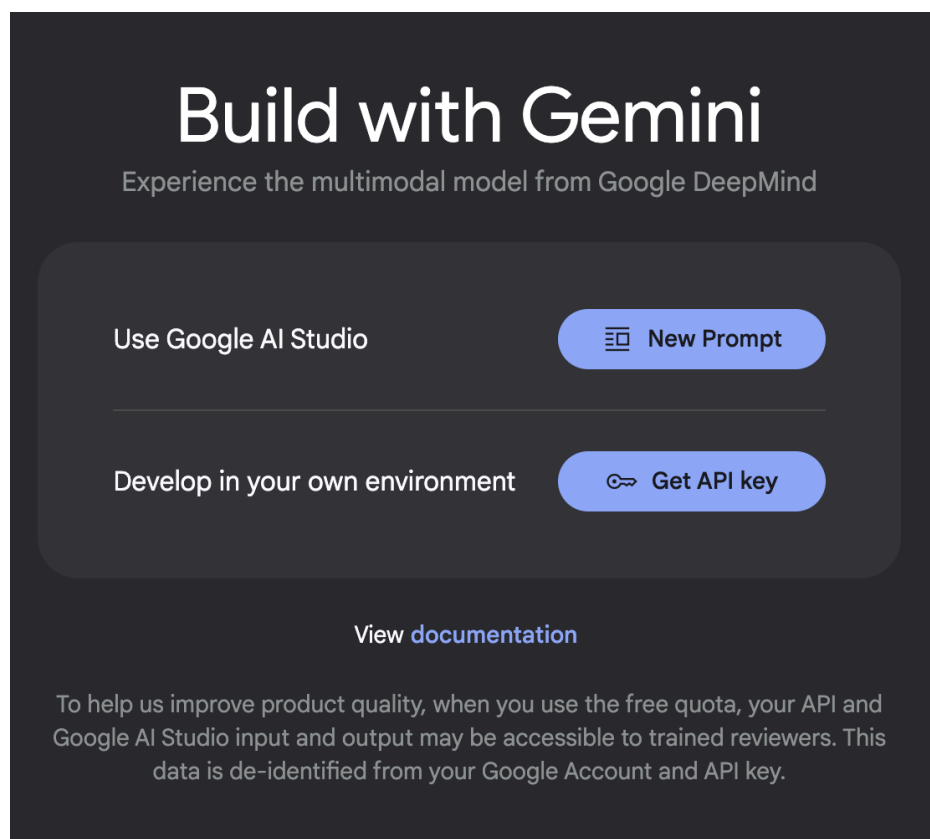


Figura 6. Pantalla inicio Google AI Studio

Debe hacer click en **"Get API Key"**

Al hacer click en **"Get API Key"**. Se cargará la pantalla "obtén una clave de API", es en esta página que usted debe hacer click en **"Crear clave de API"** (Figura 7). Esto generará una clave que usted puede utilizar para integrar cualquier programa con el servicio de Google AI Studio, el cuál es Google Gemini.

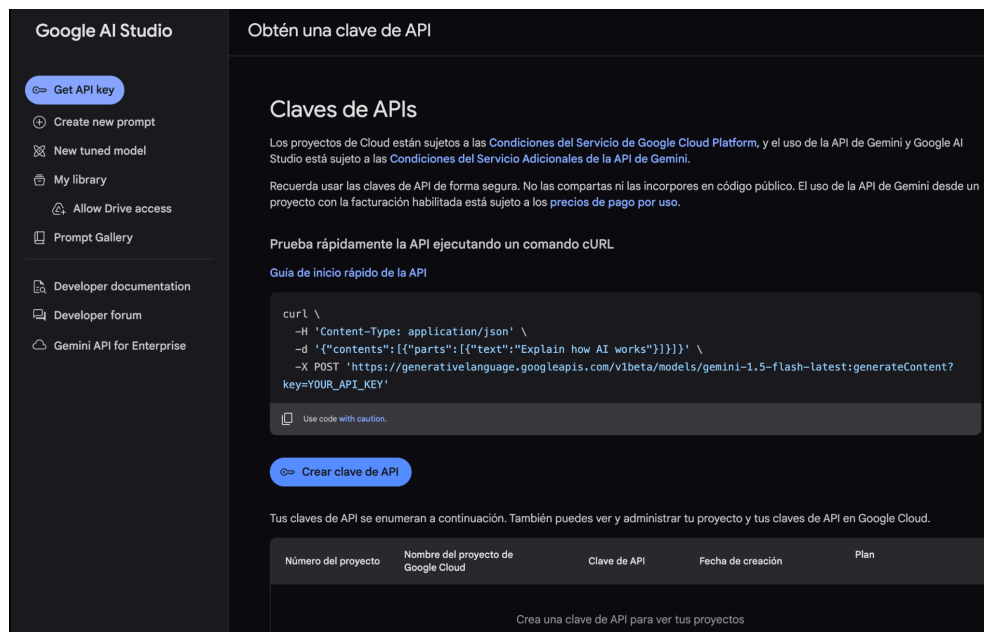


Figura 7. Generación clave API en Google AI Studio

Posteriormente aparecerá un aviso legal el cuál usted sólo debe presionar el botón **"Continuar"** (Figura 8). De igual forma se mostrará un aviso de safety setting reminder (Figura 9) el cuál debe presionar "Continuar".

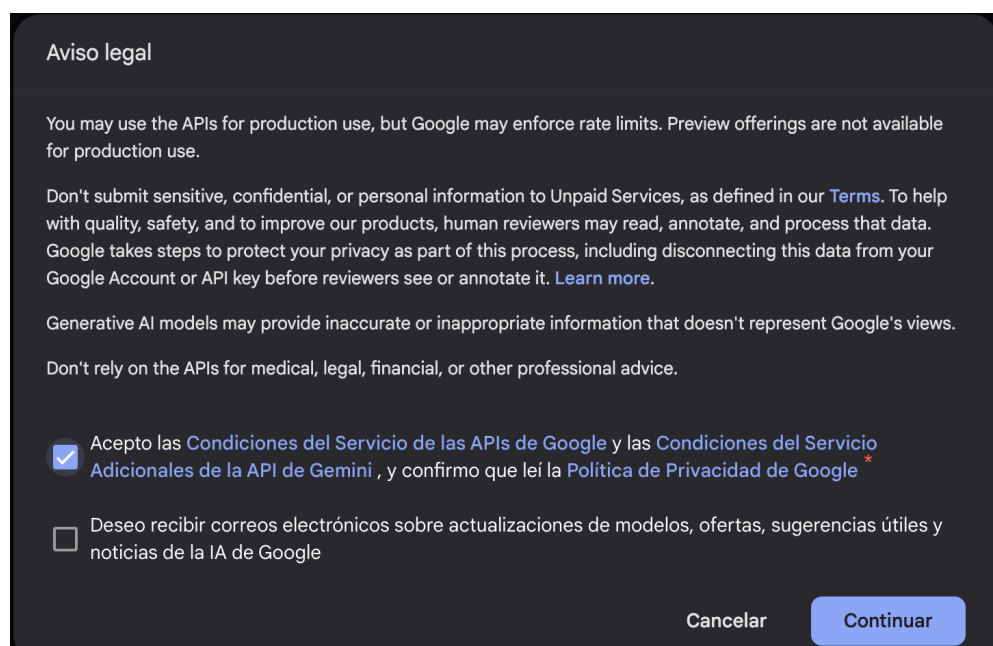


Figura 8. Aviso Legal Google AI Studio

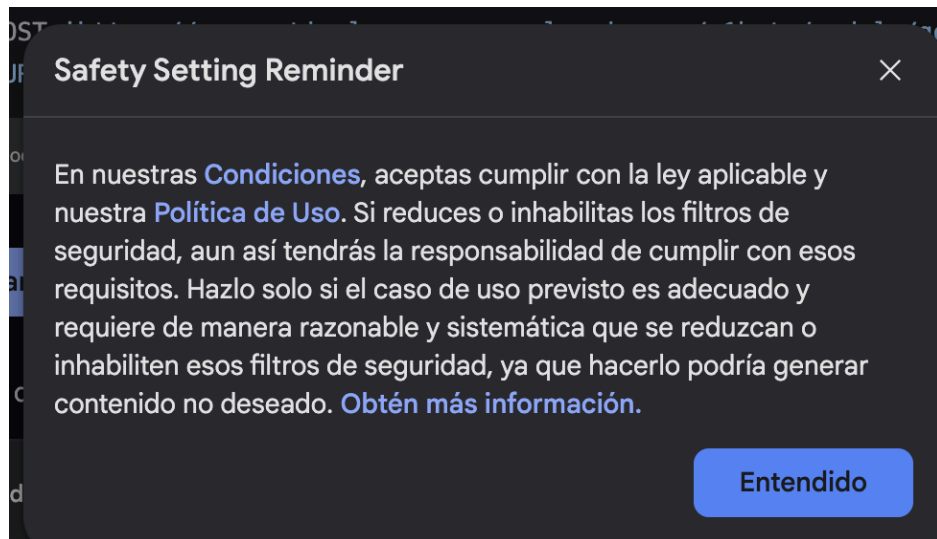


Figura 8. Aviso Legal Google AI Studio

Posteriormente aparecerá una ventana de creación de la API KEY. Es importante seleccionar la opción **"Crear clave de API en un proyecto nuevo"** (Figura 9 y Figura 10 donde el servidor comienza la creación de la API KEY).

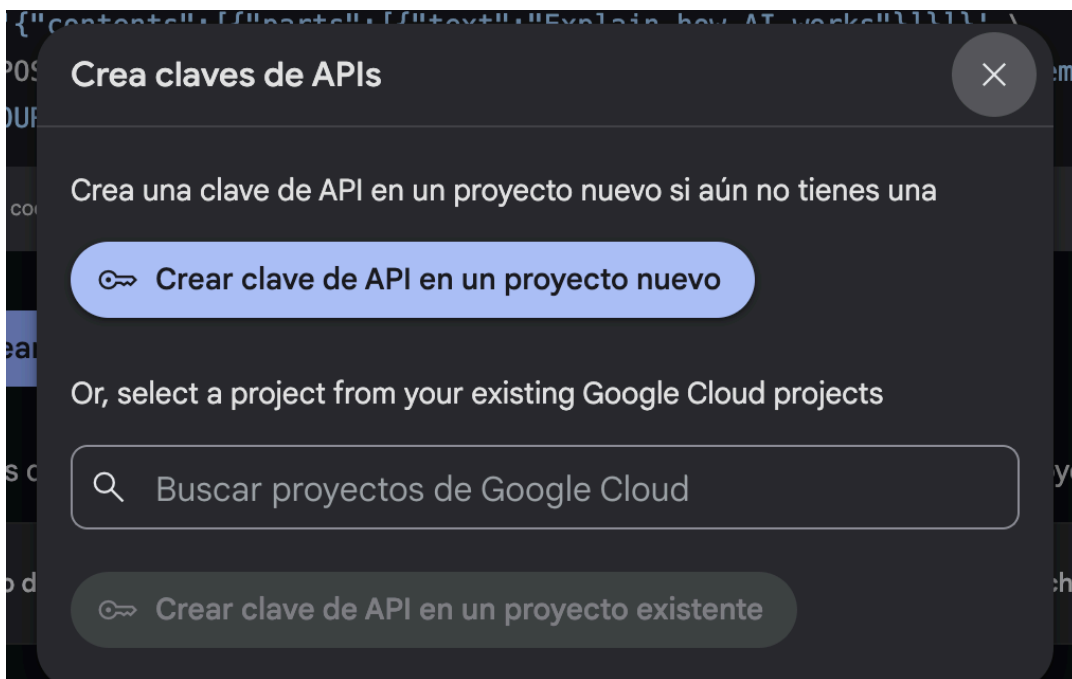


Figura 9. Crear clave de API en un proyecto nuevo

Posteriormente aparecerá una ventana de creación de la API KEY. Es importante seleccionar la opción **"Crear clave de API en un proyecto nuevo"** (Figura 9 y Figura 10 donde el servidor comienza la creación de la API KEY).

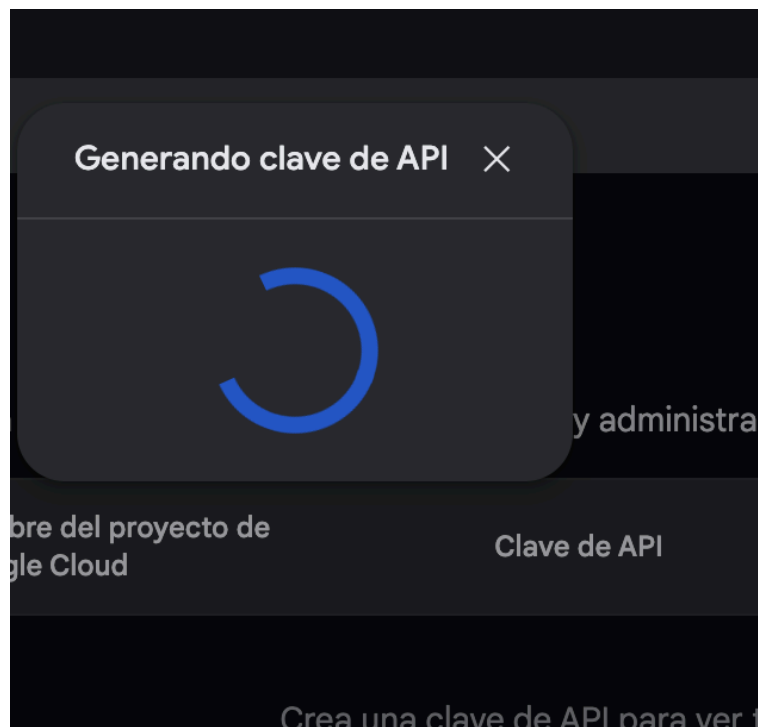


Figura 10. Estado de la generación de la API KEY

Finalmente la clave de la API (API KEY) se generará y mostrará en pantalla la clave generada. Es importante que usted copie y guarde dicha información. Esto se evidencia en la Figura 11 -por temas de privacidad la clave generada para realizar este tutorial fue marcada en rojo-. Es importante notar que la clave API KEY no puede ser compartida entre varios estudiantes, cada estudiante debe generar su propia API KEY.

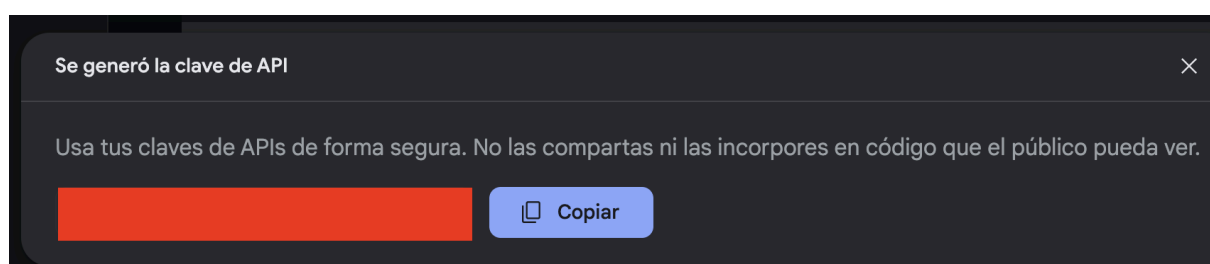


Figura 11. API KEY generada

La clave generada quedará disponible en su página inicial de la plataforma Google AI Studio (Figura 12).

Tus claves de API se enumeran a continuación. También puedes ver y administrar tu proyecto y tus claves de API en Google Cloud.


Número del proyecto	Nombre del proyecto de Google Cloud	Clave de API	Fecha de creación	Plan
...6145	Generative Language Client 	...85WI	19 sept 2024	Sin costo Configurar facturación Ver datos de uso

Figura 12. Resumen de las API KEY generadas.

Nota: usted puede generar múltiples API KEY pero no es relevante para este laboratorio.

Es responsabilidad del estudiante buscar cómo usar este servicio en cada laboratorio y cómo resolver los requerimientos solicitados en cada enunciado. En clases no se revisará ninguno de estos aspectos por lo que la investigación, consultar documentación, otros es responsabilidad del estudiante.

Comodines

Comodines: Durante el semestre cada estudiante dispondrá de **TRES comodines (uno para cada laboratorio)** que podrá utilizar **solo en el siguiente caso:**

- **Si su calificación** en los laboratorios 1, 2, 3 (Ya sea por requerimientos funcionales, no funcionales, informe) **es inferior a 4.0 y superior o igual a un 2 (es decir, que entregue un avance del proyecto).** No entregar laboratorio será calificado con un 1.0 y por tanto no tendrá derecho a usar comodín.

El comodín permitirá a el/la estudiante entregar lo necesario (requerimientos funcionales, no funcionales e/o informe) para alcanzar la nota mínima requerida (4.0) en la correspondiente entrega de laboratorio para la aprobación al final del semestre. Además, la nota máxima a la que puede optar por uso de comodín es un 4.0. **Es fundamental que lo/as estudiantes siempre entreguen un avance (dentro de los plazos establecidos para cada laboratorio) que permita alcanzar al menos una calificación igual o superior a 2.0 para poder optar al uso de comodines. Recordar que para aprobar el laboratorio de esta asignatura, todos los laboratorios deben tener una calificación igual o mayor a 4.0.**

Se recomienda no emplear los comodines como una forma de aplazar la entrega de sus laboratorios, puesto que irá acumulando trabajo hacia el final de semestre. Por lo mismo, procure trabajar en el laboratorio dentro de los plazos señalados y entregarlo aún cuando no haya podido terminarlo.

Para el caso del uso de comodín por versionamiento solo dispondrá de 2 oportunidades. Si el estudiante tiene una calificación menor a 4.0 debido al no uso o uso inadecuado de git (frecuencia, constancia, periodo), el estudiante podrá enmendar esta calificación haciendo uso efectivo de git en los siguientes laboratorios o en los anteriores. Para estos casos, el estudiante debe enviar la solicitud de aplicación de comodín por versionamiento al término del semestre a través de la plataforma habilitada para tales efectos. De ser aceptado el comodín, la nota de cierre del correspondiente laboratorio será un 4.0.

Procure realizar la autoevaluación rigurosa de cada laboratorio de manera que conozca a priori la calificación estimada que podría recibir. De esta manera se espera que tengan claridad sobre si debe o no usar comodín y en qué funcionalidades debería concentrarse para alcanzar la nota mínima. Las calificaciones de laboratorio tomarán tiempo (en el calendario Clase a Clase se presenta una fecha estimada donde debería recibir las calificaciones de cada laboratorio), por lo que se recomienda NO esperar a tener una calificación del laboratorio para recién determinar si requiere usar un comodín. Esto lo puede determinar a priori con una autoevaluación rigurosa y a conciencia. **Al respecto señalar que la evaluación de cada laboratorio está a cargo de un solo profesor, por lo que es probable que las calificaciones de un laboratorio no estén antes de la entrega del siguiente.**

Si su autoevaluación indica que su calificación hubiese sido mayor o igual a 4.0, se recomienda a lo/as estudiantes que empleen las instancias de correcciones/apelaciones de cada laboratorio antes de proceder a trabajar en el comodín. Si su autoevaluación fue a conciencia y rigurosa, pero aún así obtiene una calificación inferior a 4.0, podría tratarse de (1) un error/omisión en la revisión, (2) problemas en la implementación/ejecución que no fueron identificados por el/la estudiante, (3) problemas de versionamiento, interpretación, compilación debido a elementos que no fueron debidamente documentados y (4) aplicación incorrecta del paradigma, entre otras razones.

Finalmente, en caso de requerir el uso de comodín, podrá hacer entrega del correspondiente laboratorio corregido al final del semestre en las fechas y canal estipulados para estos efectos. La fecha específica de entrega de comodines se encuentran señalados en el Clase a Clase, mientras que los espacios de entrega para cada comodín estarán disponibles en CampusVirtual.

Instrucciones generales para la entrega de TODOS los laboratorios

Versión 1.0

(Cambios menores pueden incorporarse en futuras versiones a fin de aclarar o corregir errores)

(Sus dudas las puede expresar en este mismo enunciado, incluso puede responder a preguntas de compañeros en caso de que conozca la respuesta)

Entregables: Archivo ZIP con el siguiente nombre de archivo: labN_rut_ApellidoPaterno.zip (ej: lab1_12123456_Perez.zip). **Notar que no incluye dígito verificador.** Dentro del archivo se debe incluir:

1. Informe en formato PDF.
2. Carpeta con código fuente.
3. **Archivo leeme.txt** para cualquier instrucción especial para ejecución u otro, según aplique.
4. **Autoevaluación.txt** donde debe incluir una lista completa de todos los requerimientos funcionales y no funcionales señalando una evaluación para cada uno según la siguiente escala:
 - a. 0: No realizado.
 - b. 0.25: Implementación con problemas mayores (funciona 25% de las veces o no funciona)
 - c. 0.5: Implementación con funcionamiento irregular (funciona 50% de las veces)
 - d. 0.75: Implementación con problemas menores (funciona 75% de las veces)
 - e. 1: Implementación completa sin problemas (funciona 100% de las veces)
5. **Archivo repositorio.txt** que incluya la URL de su repositorio privado en GitHub² el que deberá además compartirlo con los profesores y ayudantes al momento de la evaluación a la cuenta **paradigmasdiinf** de GitHub (no lo comparta antes ya que las invitaciones expiran después de un tiempo). **El nombre final del repositorio a compartir debe ser el mismo del entregable en Campus Virtual.**

El nombre de todos los archivos con código fuente, informes, autoevaluación, etc. debe seguir el siguiente formato: **NombreArchivo_rut_Apellidos.extension** (ej: labDobble_12123456_PerezPeña.rkt; funciones_12123456_PerezPeña.rkt)

Se aplicarán descuentos en caso de no incluir estos datos e inclusive la nota mínima si el programa no puede pasar por los procedimientos automáticos de revisión por no respetar esta instrucción.

Informe del proyecto: Extensión no debe superar las **7 planas** de contenido (se excluyen: portada, índice, referencias en formato APA, anexos), para más detalle de requerimientos del informe [ver descripción en nuestro espacio en Campus Virtual](#). El informe debe incluir introducción, descripción breve del problema, análisis del problema, diseño de la solución (esquematizada y explicada brevemente), consideraciones de implementación (ej:

² Ver tutorial de Git en Campus Virtual donde se explica como activar el Student Pack para poder crear repositorios privados de forma gratuita.

algoritmos, bibliotecas), **instrucciones con ejemplos claros de uso**, resultados obtenidos, evaluación completa y conclusiones.

Repositorio en Git del Proyecto: El laboratorio deberá ser desarrollado incluyendo el sistema de control de versiones Git y utilizando GitHub como servidor remoto. **El repositorio debe ser privado durante TODA la elaboración del proyecto.** Cualquier inconsistencia entre lo entregado en Campus Virtual y la versión final del repositorio será evaluado con nota mínima. **Para el proceso de revisión se considerará la versión subida a CampusVirtual.**

IMPORTANTE: el nombre de cada repositorio en cada laboratorio debe contener la siguiente información:

LabX_Rut_NombreApellidos

Ejemplo, el nombre del repositorio 1: Lab1_20123123K_JuanArayaPerez

En caso de que su repositorio no tenga la información previamente descrita, no se podrá evaluar su repositorio, en consecuencia obtendrá la nota mínima.

Integridad: Lo/as estudiantes deben desarrollar los laboratorios de forma individual. Si bien lo/as estudiantes pueden apoyarse y discutir aspectos del proyecto, el diseño e implementación del mismo, la solución construida se debe producir de manera individual. No se pueden compartir diseños o implementaciones, tampoco se pueden obtener de terceros. Como parte de la revisión existen procedimientos de revisión de copia/plagio tanto a nivel de código como en los informes. **Al detectarse plagio, el trabajo en cuestión no será revisado por no contar con evidencia concreta sobre el resultado de aprendizaje esperado para el correspondiente laboratorio. Por lo anterior, será calificado con un 1.0. El/la estudiante tampoco podrá hacer uso de comodín en este caso ya que es el equivalente a no haber entregado dicho laboratorio. Por último se iniciará investigación sumaria a través de la Facultad de Ingeniería con el objeto de determinar las sanciones correspondientes.**

Evaluación: El Informe (Inf), requerimientos funcionales (RF), requerimientos no funcionales (RNF), ejecución (Eje) se evalúan por separado. La nota final de cada laboratorio (NL) se calcula de la siguiente forma considerando sólo la calificación base a partir del cumplimiento de los RF y RNF obligatorios.

*if (Inf >= 4.0 && (RF + 1.0) >= 4.0 && (RNF + 1.0) >= 4.0) then
NL = 0.1 · Inf + 0.7 · (RF + 1.0) + 0.2 · (RNF + 3.0)
else
NL = min(Inf, RF + 1.0, RNF + 1.0)*

;Respecto de este caso, si el estudiante entrega el laboratorio
;y algunas de sus partes (informe, RF, RNF) es 1.0 por no entrega
;**será calificado con un 2.0** para que pueda optar a comodín.
;Por lo anterior, procure siempre hacer una entrega.

Acceso a los laboratorios específicos

[Laboratorio 1 \(Funcional - Scheme\)](#)

[Laboratorio 2 \(Lógico - Prolog\)](#)

[Laboratorio 3 \(Orientado a Objetos - Java\)](#)