

PARADIGMAS DE PROGRAMACIÓN

PROYECTO SEMESTRAL DE LABORATORIO

Versión Preliminar - actualizada al 30/08/2023

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 un sistema para la creación, despliegue y administración de chatbots **simplificado**.

Los estudiantes desarrollarán soluciones a partir de los conceptos y principios de distintos paradigmas de programación. Estas soluciones permitirán a un usuario realizar distintas operaciones (ej: crear, vincular, interactuar, etc.) en un sistema centrado en la operación con chatbots. El software en su versión final (Laboratorio 3) tendrá una interfaz de usuario a través de la cual se podrán realizar acciones sobre el sistema de chatbots en base a línea de comandos. Opcionalmente podrán implementar una interfaz gráfica que imita a la que se usan en plataformas de ecommerce.

Varios aspectos del software quedarán abiertos a la creatividad de cada estudiante, sin embargo se debe cumplir con requerimientos mínimos obligatorios, los cuales se especifican en las siguientes secciones.

A continuación se listan de forma general características del software que serán abordadas de forma obligatoria o complementaria durante el semestre en los distintos proyectos. Los detalles para cada laboratorio se listan en enunciados específicos cuyos enlaces de acceso se presentan más abajo. **Para todo los efectos, este documento contiene los lineamientos generales que aplican a todos los laboratorios y por tanto debe ser consultado en cada laboratorio.**

Existen distintos tipos de chatbots. Para los efectos de este laboratorio se trabajará con chatbots de ITR (Respuesta de Interacción a Texto). Este tipo de chatbots se caracteriza por ser muy estructurado, por responder a preguntas en base a opciones específicas. Las opciones pueden estar dadas por palabras clave con sinónimos o de manera más simplificada, con números o letras que permitan a un usuario escoger entre un listado de preguntas. Por otro lado, fuera del reconocimiento exacto de palabras clave, este tipo de chatbots no realizan un procesamiento elaborado del lenguaje natural. Tampoco involucran inteligencia artificial. Un ejemplo de este tipo de chat se ilustra en la Figura 1 donde el chat

primero ofrece un mensaje de bienvenida y a continuación opciones desde las cuales el usuario puede seleccionar la se ajuste a su necesidad. De esta manera se produce un “diálogo” estructurado entre usuario y sistema.

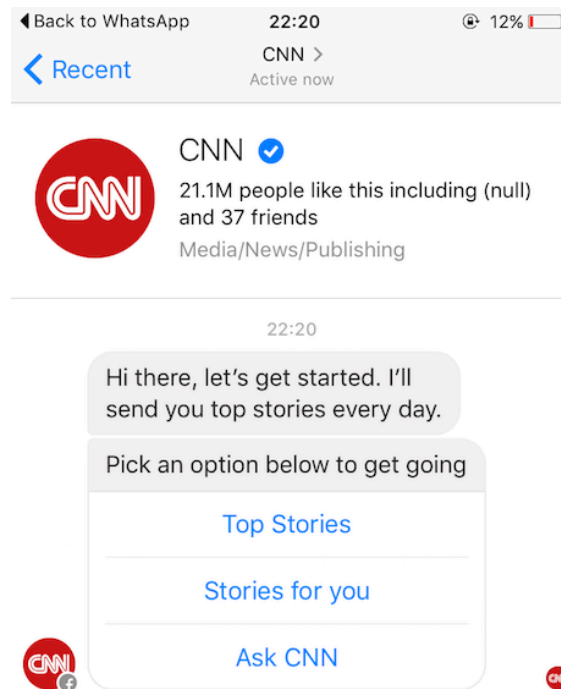


Figura 1. Ejemplo de Chatbot Estructurado

El ambiente que se abordará en el marco de este laboratorio permitirá crear distintos chatbots. Al ser cada chatbot de propósito específico, estos se podrán entrelazar de manera que opciones de un chatbot puedan derivar la atención de un usuario a otro chatbot (Figura 2). De esta forma los tipos de preguntas y respuestas asociadas están definidas a priori.

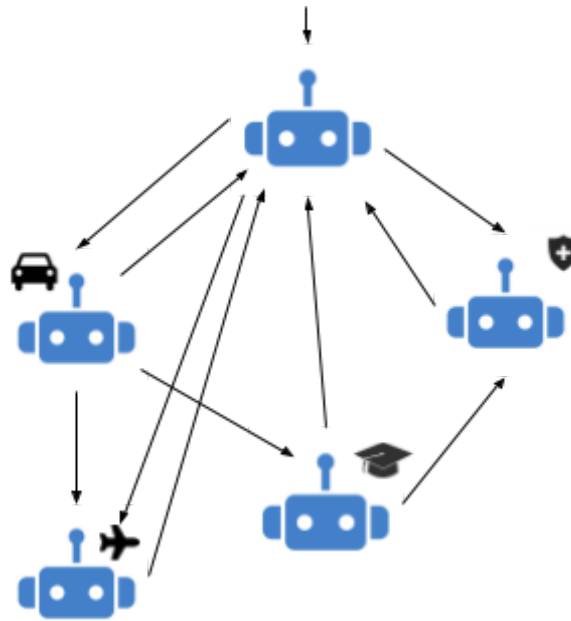


Figura 2. Interconexión de chatbots

En la Figura 2 se muestra un chatbot inicial desde donde se producen diálogos que pueden ir dirigiendo al usuario a otros chatbots. A modo de ejemplo, el usuario podría requerir servicios de arrendo de autos y desde ahí, mediante una opción del diálogo, se podría enlazar con un chatbot de agencia de viajes aéreos para luego volver al chatbot inicial.

Un ejemplo más elaborado del chatbot se ilustra en la Figura 3.

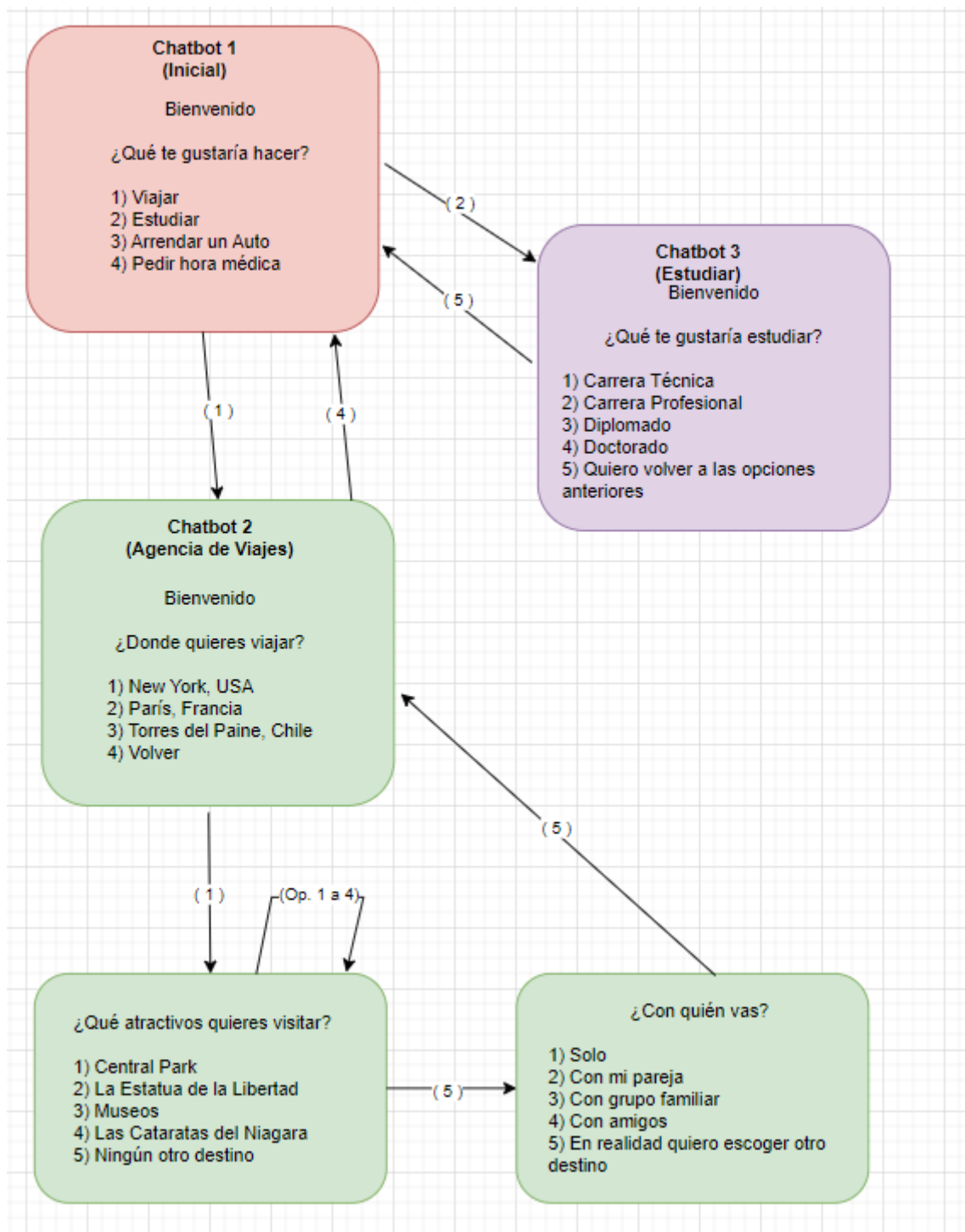


Figura 3. Vista parcial de chatbots e interacciones entre éstos.

Cabe señalar que las interacciones con el chatbot pueden ser a través de números o letras que denotan las opciones o bien a través de palabras claves que se vinculan a la opción. A modo de ejemplo, para la opción “Viajar” en el primer diálogo del Chatbot 2, se podría interactuar con textos como “Viajar”, “viajar”, “turistear” y “travel”, entre otros.

El ambiente que usted implementará debe permitir a un usuario las siguientes operaciones:

- 1) Crear un ambiente contenedor de chatbots

- 2) Crear un chatbot
 - a) Identificar chatbot
 - b) Añadir preguntas
 - c) Añadir opciones
 - d) Especificar enlaces para el flujo de interacción dentro del chatbot y hacia otros chatbots
- 3) Interactuar con el chatbot
 - a) Mediante opciones
 - b) Mediante textos considerando match exacto con palabras clave
- 4) Ofrecer una síntesis de las interacciones con el chatbot. A modo de ejemplo, en el caso del flujo de la Figura 3, si un usuario escoge viajar, luego especifica su destino, los atractivos que quiere visitar, con quien viajará, etc. Al finalizar el flujo, el chatbot debería concluir con un mensaje del estilo. “Su viaje ha sido agendado de acuerdo a los siguientes datos: Destino, Atractivos, Grupo, etc.”

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 al menos entrego algo).** 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 y/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 a conciencia.

Si su autoevaluación indica que su calificación hubiese sido mayor o igual a 4.0, **se recomienda a lo/as estudiantes empleen las instancias de correcciones 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 GitHub¹ 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 **5 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:

¹ 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.**

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

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

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

[Laboratorio 3 \(POO - Java o C#\)](#)