

## TAREA PROGRAMADA 2

### Introducción

La navidad es una época muy agradable para mucha gente, porque comen pavo y tamales y todo eso. Para Santa, sin embargo, es una época complicada, por el hecho de que hay muchos niños a los cuales les tiene que dar regalos, y a veces esa es una tarea complicada, porque hay aproximadamente unos 3 mil millones de niños en el mundo, y no todos se portan bien.

Se han dado casos extremos en donde niños bien portados que se aprenden las tablas de multiplicar han recibido solamente medias de golfista y ropa interior de Bob Esponja, mientras que los niños malos han recibido LEGOs y juegos de PS3.

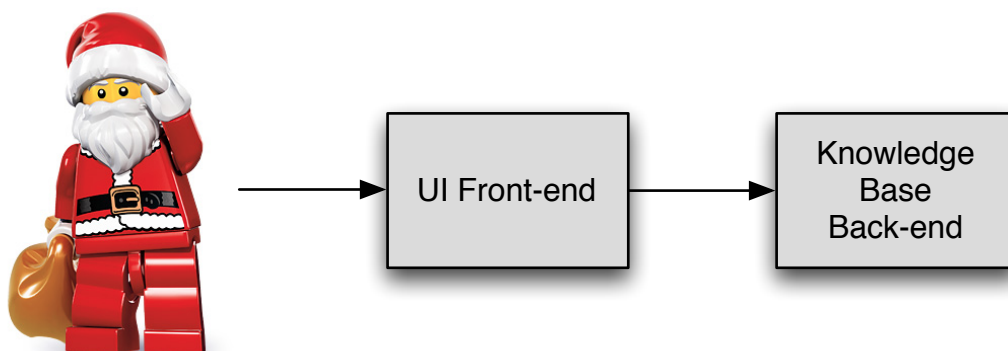
Es por eso que a usted, como Jefe de la división de Desarrollo de Software de Polo Norte Inc., se le ha encargado la tarea de desarrollar un sistema de consulta de información de regalos, para simplificar las tareas de Santa. La idea es que él o alguno de sus duendes puedan usar el sistema para ingresar la información de los niños, y posteriormente hacer consultas acerca de las características de esos niños, para saber cómo se portaron durante el año, y con base en eso escoger el regalo que les toca.

### Descripción del programa

El sistema tendrá dos modalidades principales: mantenimiento de datos y consulta. En la modalidad de mantenimiento de datos, los encargados del taller de Santa podrán ingresar los datos de todos los niños del mundo. La información de los niños que se almacenará será la siguiente: nombre del niño, edad, país, lista de buenas acciones, lista de malas acciones, lista de regalos deseados (wishlist), presupuesto de Santa para ese niño. Por otro lado, se tendrá la información de regalos o juguetes. Para cada tipo de juguete (ejemplos de juguetes podrán ser LEGO, peluche, carro a control remoto, etc.), se tendrá la siguiente información: nombre, marca, precio, edad necesaria para usarla (hay juguetes que tienen edades mínimas),

En la modalidad de consulta, los encargados del taller podrán hacer consultas con base en ciertos atributos de niños y juguetes, y el sistema deberá mostrar la información que cumpla con los atributos ingresados.

El sistema estará estructurado en dos componentes: Front-end y Back-end.



El Front-end será el componente encargado de manejar toda la interacción con el usuario, incluyendo entrada y salida de datos. Adicionalmente, se comunicará con el Back-end para las tareas relacionadas con mantenimiento de datos y consulta.

El Back-end será el encargado de manejar la base de conocimientos de recetas, y de responder las consultas hechas por el usuario (aunque de manera indirecta, ya que el Front-end es el que realmente maneja la interacción con el usuario, y hace a su vez las consultas al Back-end).

El Back-end tendrá dos funcionalidades principales: manejar la base de conocimientos y responder las consultas.

La base de conocimientos será una serie de declaraciones en Prolog que definan todos los atributos de los niños y los juguetes mencionados anteriormente en el documento. El Back-end deberá actualizar la base de conocimientos usando los datos que el usuario ingrese al sistema por medio del Front-end.

La estructura de la base de datos podrá ser definida por los estudiantes, no tiene que tener un formato particular. Sin embargo, en la documentación se deberán explicar las decisiones de diseño detrás de la estructura elegida..

La lista de consultas que deberá tener la aplicación es la siguiente:

- Lista de niños malos y niños buenos (buenos = más acciones buenas que malas, malos = más acciones malas que buenas)
- Lista de niños que tienen una acción buena en particular (el usuario puede escribir una acción y el sistema lo busca en la lista de acciones buenas de los niños)
- Lista de niños que tienen una acción mala en particular (el usuario puede escribir una acción y el sistema lo busca en la lista de acciones malas de los niños)
- Lista de regalos más solicitados por los niños, ordenados por popularidad
- Consulta niño-regalos: el usuario digita el nombre de un niño, y el sistema despliega la información de regalos que quiere ese niño, con sus respectivos datos (precio, marca). Si el niño tiene más acciones malas que buenas, el sistema deberá indicar que al niño se le tienen que dar medias de golfista
- Consulta niño-regalos posibles: igual que la consulta anterior, pero en este caso, se filtran solo los regalos que sean válidos para la edad del niño (usando la edad del niño y la edad mínima para el juguete), y filtrando los juguetes que están fuera del presupuesto para ese niño (en lo que el precio del regalo es mayor al presupuesto asignado por Santa para ese niño). Si el niño tiene más acciones malas que buenas, el sistema deberá indicar que al niño se le tienen que dar medias de golfista
- Presupuesto total: deberán mostrar una lista de todos los regalos que se les deben dar a los niños buenos, tomando en cuenta las consideraciones de edad y de precio especificadas en el punto anterior. Además, deberán mostrar el monto total que deberá invertir Santa para poder comprar todos los regalos.

El modo de funcionamiento es el siguiente: el usuario digita una consulta en el Front-end, y posteriormente, este deberá comunicarse con el Back-end, para que realice las consultas pertinentes a la base de conocimientos, y devolver la información al Front-end, para que sea desplegada al usuario

correctamente. En caso de que no haya ninguna receta con las características digitadas, el sistema deberá mostrar un mensaje que lo indique.

## Aspectos técnicos

El Back-end deberá estar escrito en el lenguaje de programación Prolog.

Con respecto al Front-end, deberá estar escrito en un lenguaje de programación distinto a Prolog, el cual podrá ser elegido por cada grupo (si el grupo usa un lenguaje distinto a Python, tendrá un 10% adicional en la nota). El Front-end deberá tener una interfaz gráfica de usuario (no se permite el uso de modo consola).

El proyecto completo deberá funcionar en el sistema operativo Linux, por lo que es importante que los grupos tomen eso en consideración a la hora de elegir el lenguaje de programación para el Front-end.

## Aspectos a investigar

Los estudiantes deberán investigar la forma en la que se realizará la conexión entre Prolog y el lenguaje de programación elegido. Podrán hacer uso de alguna librería externa para hacer la conexión.

## Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos. Para la documentación interna, deberán incluir comentarios descriptivos para cada función o predicado, que permita entender con claridad el propósito del código.

La documentación externa deberá incluir:

- Tabla de contenidos
- Descripción del problema
- Diseño del programa: decisiones de diseño, algoritmos usados, descripción de principales predicados, lenguaje de programación escogido para el componente de UI, y las razones detrás de dicha escogencia. Es importante que en esta sección documenten bien todas las decisiones tomadas a la hora de desarrollar la tarea.
- Librerías usadas: librerías de GUI usadas, librerías de interconexión entre prolog y el lenguaje escogido, razones de escogencia de esa librería
- Análisis de resultados: descripción de las tareas que se completaron y de las tareas que no se lograron completar; además de algunas propuestas de solución para los problemas que no se pudieron resolver
- Manual de usuario: instrucciones de uso de modalidad mantenimiento y consulta, instrucciones de compilación (en caso de requerir librerías adicionales, deberán detallar todos los pasos que se deben ejecutar para compilar y ejecutar el código, además de incluir los archivos de las librerías necesarios dentro del código fuente que entregarán)
- Conclusión personal

## Evaluación

Documentación interna	2%
Documentación externa	8%
Funcionalidad modo mantenimiento	15%
Funcionalidad modo consulta	15%
Interconexión prolog-lenguaje programación	20%
Manejo de interfaz gráfica de usuario	10%
Revisión	30%
Escogencia de lenguaje para Front-end	10% adicional

## Aspectos administrativos

- La tarea vale un 10% de la nota del curso
- La tarea se hará en grupos de 2-3 personas.
- Fecha de entrega: Martes 14 de mayo, 3 p.m. No se aceptan tareas entregadas después de esa fecha y hora, sin excepciones.
- Los grupos deberán subir el código y la documentación de sus respectivas tareas a un repositorio en Github, de manera que el profesor pueda ver las contribuciones que las diferentes personas hacen al proyecto. La idea es que apenas empiecen a desarrollar la tarea, suban las contribuciones al repositorio, y no esperar a tener todo el código listo para subirlo.
- Deberán enviar un correo a [andreifu@gmail.com](mailto:andreifu@gmail.com), con copia a [evelyn.madriz@gmail.com](mailto:evelyn.madriz@gmail.com), en donde indiquen el url del repositorio de Github en donde se encuentra el código y la documentación de la tarea. El asunto del correo enviado tendrá el siguiente formato: **TI-3404 TP2-nombrea1-nombre2**. Las tareas que no sean entregadas por medio de Github tendrán nota cero.
- Las tareas deberán ser revisadas con el profesor o el asistente. Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente. La nota de la revisión es individual, el resto de la nota es grupal.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea, se aplicarán **todas** las medidas correspondientes, según el reglamento del TEC, incluyendo una carta al expediente.