

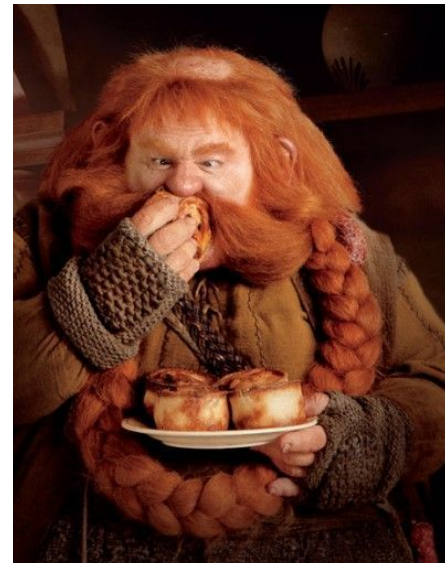
Tarea programada 2

Introducción

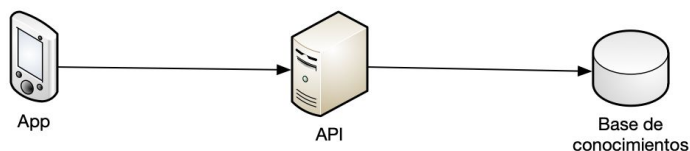
Bombur es el más gordo y glotón de todos los Enanos de Tierra Media, por lo que siempre pasa inventando nuevas recetas con base en los ingredientes que va encontrando en sus aventuras en lugares como La Comarca, Rivendell, Mirkwood, Erebor y Bree.

Ahora, guardar todas esas recetas en manuscritos es problemático, porque se dañan fácilmente, y además no es sencillo encontrar información de las recetas, por lo que Bombur gasta mucho tiempo en eso.

Para ayudarle a Bombur a administrar sus recetas, vamos a desarrollar un sistema basado en una aplicación móvil, un API, y una base de conocimientos y un mecanismo de inferencia para consultas.



Descripción del programa



App

- Aplicación móvil escrita en Android nativo con las siguientes funcionalidades:
 - Crear cuenta, loguearse y desloguearse
 - Visualizar recetas en una lista
 - Si se da click en una receta, deben mostrar los siguientes datos: nombre, tipo de comida, lista de ingredientes (pueden ser n ingredientes), instrucciones (pueden ser n pasos), fotos (pueden ser n fotos por receta).
 - Agregar nuevas recetas: se debe de poder agregar nombre, tipo de comida, lista de ingredientes, instrucciones y fotos.
 - Buscar recetas por nombre, tipo, o ingredientes

API

- Backend escrito en Ruby o Python, debe correr sobre AWS o a Heroku
- El API debe manejar autenticación por medio de tokens (o sea, usuarios no autenticados no deben poder usar el API)
- Endpoints:
 - Crear cuenta
 - Login usuario
 - Obtener lista de recetas
 - Obtener detalle de una receta
 - Agregar una receta
 - Buscar recetas por nombre
 - Buscar recetas por tipo de comida
 - Buscar recetas por ingrediente (devuelve recetas con ese ingrediente)

Base de conocimientos

- Base de conocimientos y mecanismo de inferencia escritos en Prolog
- La base de conocimientos será una serie de declaraciones en Prolog que defina todos los atributos de las recetas. El API deberá comunicarse con la base de conocimientos para agregar y consultar recetas, usando los datos que el usuario ingrese al sistema por medio de app.

Aspectos técnicos

- La información de usuarios sí la pueden guardar en bases de datos relacionales del lado del API
- Todos los datos de recetas deben guardarse en Prolog, en la base de conocimientos
- El app no interactúa con la base de conocimientos, lo hace a través del AP.
- Las consultas para las búsquedas tienen que ser resueltas a nivel de Prolog, no a nivel de lógica del API. El API solamente toma los datos y los manda al app.

Aspectos administrativos

- La tarea vale un 30% de la nota del curso
- La tarea se hará en grupos de 3 personas máximo.
- Fecha de entrega: Miércoles 5 de Junio, 7:30 a.m. No se aceptan tareas entregadas después de esa fecha y hora. Las revisiones serán a partir de ese mismo día, por lo que no hay extensión de tiempo.
- 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.tec@gmail.com, con copia a gaboq98@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: IC-4700 TP2-nombrea1-nombre2-nombre3. 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.
- El código entregado debe ser 100% original. 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.

Evaluación

La siguiente rúbrica especifica cómo se calificarán los diferentes elementos que forman parte de la evaluación.

Criterio	%	Excelente (100%)	Adecuado (70%)	Regular (30%)	Deficiente (0%)
Documentación	5%	Usan buena redacción. La documentación está completa y bien elaborada	Errores de redacción menores. La documentación falla en detallar aspectos menores	Errores de redacción frecuentes. La documentación no tiene incluye aspectos críticos	No entregan documentación.
Registro, login y logout	5%	Se puede crear un nuevo usuario desde cero. Se puede hacer login. Se puede hacer logout. Los usuarios se guardan a nivel de base de datos en el API.	Alguna de las funcionalidades tiene un error menor	Hay un error crítico o dos o más errores menores en alguna de las funcionalidades	Hay dos errores críticos, o tres o más errores menores en alguna de las funcionalidades
Autenticación de API	5%	El API maneja autenticación usando tokens	N/A	N/A	No manejan autenticación por tokens, o tiene un error.
Listar recetas	10%	Se pueden visualizar todas las recetas en una lista. Las listas se cargan desde el API. El API carga las recetas desde la base de conocimientos en Prolog	Alguna de las funcionalidades tiene un error menor	Hay un error crítico o dos o más errores menores en alguna de las funcionalidades	Hay dos errores críticos, o tres o más errores menores en alguna de las funcionalidades
Ver detalle de recetas	10%	Se pueden ver todos los detalles de recetas: nombre, tipo, ingredientes, pasos y fotos, y todos los datos se cargan del API/Base de conocimientos	Al menos uno de los datos no se muestra, o no se cargan desde el API/Base de conocimientos	Al menos dos de los datos no se pueden ver, o no se cargan desde el API/Base de conocimientos	Tres o más datos no se pueden ver, o no se cargan desde el API/Base de conocimientos

Agregar recetas	25%	Se pueden agregar recetas con todos los datos. Las recetas se guardan como hechos en la base de conocimientos en Prolog	Al menos uno de los datos no se puede guardar.	Al menos dos de los datos no se pueden guardar.	Tres o más datos no se pueden guardar, o guardan los datos pero no los guardan en Prolog.
Buscar recetas	30%	Se pueden buscar recetas por nombre, tipo de comida, o ingredientes	Al menos una de las búsquedas no funciona	Al menos dos de las búsquedas no funcionan	Tres o más búsquedas no funcionan, o hacen la búsqueda a nivel manual desde el API
Desempeño en revisión de tarea	10%	Responden adecuadamente a cada una de las preguntas hechas durante la revisión. Demuestran conocimiento avanzado de la materia y de la solución entregada.	Muestran conocimiento medio de la materia y de la solución entregada.	Muestran conocimiento muy básico de la materia. No responden correctamente algunas de las preguntas.	No asisten a la revisión.