

Laboratorio 1: Entrega parcial.

Profesor teoría: Juan Iturbe A. (juan.iturbe@usach.cl)

Profesor laboratorio: Joaquín Donoso O. (joaquin.olivares@usach.cl)

Contexto

En la actualidad, el software es una pieza fundamental en los procesos de transformación digital en que se ven enfrentadas las organizaciones de diversa índole de nuestra sociedad. Estas herramientas, como cualquier otro activo, deben ser altamente eficientes y con un fuerte enfoque en la calidad.

Según la norma ISO 25.010, que refiere a la calidad en el producto software, uno de los pilares que indican una buena salud de este atributo y que es de nuestro especial interés, es la seguridad.

Pese a la transparencia que significan para los usuarios finales, los procesos de desarrollo de software son el motor de una industria que día a día dan soporte constante a la era digital en que nos encontramos inmersos.

1

Ambiente de desarrollo

- Cuenta en gitlab.com.
- Git.
- JDK 17.
- IDE de su preferencia. Se recomienda Spring Tools Suite.
- Lombok (elimina warnings del proyecto base).
- Postman.
- Docker (no es necesario aún).
- Proyecto base debe ser descargado o realizar un fork desde:
<https://gitlab.com/usach4/202402-lab.git>



Instrucciones

- Objetivo: Conocer el uso de las herramientas que se utilizan en la elaboración de un software a pequeña escala.
- **Entrega parcial de la experiencia N°1, corresponde al 40% del total de la nota. Enunciado con la actividad restante estará disponible el día lunes 30/09 después de clases.**
- Lograr configurar el ambiente de desarrollo y posterior entrega en un repositorio es parte de la evaluación.
- Experiencia debe ser desarrollada en el proyecto base entregado y con el stack tecnológico detallado en apartado “Ambiente de desarrollo”.
- Debe ser desarrollado por los mismos grupos asignados en las clases de teoría. Comunicarse de manera efectiva es parte de nuestro quehacer.
- Cualquier copia en código entre grupos o estudiantes será calificada con nota mínima.
- El programa debe ser robusto. Si no compila, tiene nota mínima.
- Dudas serán contestadas durante la clase de laboratorio o ambiente virtual hasta el 04/10.
- Fecha de entrega: Domingo 06/10 a las 23.55. Un integrante del grupo debe subir el código a un repositorio propio y privado en gitlab.com y conceder permisos de “Mantainer” a ambos profesores. Subir en otra plataforma es equivalente a no entregar el laboratorio, se recomienda crear el repositorio a la brevedad y hacer pruebas. Fork sería una alternativa.

2

Problemática

Una clínica veterinaria ha decidido implementar un sistema de apoyo a su gestión diaria que cuenta un módulo que permite gestionar las mascotas que asisten a la consulta. El sistema además cuenta con login y manejo de cuentas de usuarios para perfilar el acceso.

La aplicación carece de otras funcionalidades importantes para el correcto soporte a la operación, como venta de medicamentos, productos y el registro de atenciones.

Un producto tiene un nombre, una descripción, un costo, un precio de venta y una categoría (medicamentos, alimentos, accesorios y otros productos).

Una atención tiene asociado un profesional veterinario, una mascota y un horario.



Requerimientos

- Se debe generar el módulo “Productos”. Este módulo es similar en términos arquitectónicos al módulo existente de “Mascotas”. Se recomienda revisar cómo funciona ese módulo antes de programar (entities, model, repository, services y controllers).
- El módulo debe poder:
 - Crear un producto, con los parámetros necesarios.
 - Modificar un producto, dado un id y los parámetros a modificar
 - Eliminar un producto dado un id
 - Listar los productos por nombre del producto
- Se debe generar el módulo “Atenciones”, el cual debe poder procesar:
 - Crear una atención con los parámetros necesarios (id usuario, id mascota y horario)
 - Modificar una atención, dado un id y los parámetros a modificar
 - Eliminar una atención dado un id.
 - Listar las atenciones por id de profesional.
- Se debe utilizar de manera correcta los verbos HTTP.
- La collection Postman que se encuentra en el repositorio, debe venir correctamente actualizada para probar la nueva funcionalidad.



Bonificaciones y penalizaciones

- Entrega temprana hasta el jueves 03/10 hasta las 23.55 será bonificada con 0.5 puntos para todo el grupo.
- Documentar el código con Javadoc en los nuevos módulos, tendrá una bonificación de 0.5 puntos para todo el grupo.
- Entregar historias de usuario de las 8 funcionalidades pedidas con el formato “Como[...], Quiero[...], Para[...]” suma 0.5 puntos para todo el grupo.
- Entregar una matriz con los permisos de la aplicación dado un perfil, suma 0.5 puntos para la **siguiente entrega** (Considere los perfiles “Vendedor”, “Administrador” y “Veterinario”)
- Bonificaciones acumuladas en clases hasta el 30/09 serán sumadas a esta entrega.
- Entrega tardía cancela todas las bonificaciones conseguidas. Cada día de atraso disminuye en 2 puntos la calificación máxima.
- Si la nota es:
 - < 4.0 , se cancelan todas las bonificaciones.
 - ≥ 4.0 y < 5.0 , la bonificación disminuye al 50%.
 - ≥ 5.0 aplica bonificación completa.
- Máximo de bonificaciones no pueden superar 1.5 puntos. El excedente queda para la siguiente experiencia, si nota final es mayor o igual a 5.5.

