

Control *n*°2

1. Para la primera parte del control, deben elegir una aplicación de acuerdo a la siguiente división:

- **Aguilar – Escobar**: una aplicación de comercio en línea, como Mercado Libre, Falabella, Amazon, etc.¹.
- **Gaete – Retamal**: una aplicación educacional, como Duolingo, Udemey, Kahn Academy, edX, Coursera, etc.
- **Reyes – Zúñiga**: una aplicación de entretenimiento, como Netflix, Amazon Prime, algún juego, etc.

Indique claramente la aplicación seleccionada en su respuesta. Para la aplicación seleccionada:

- a) **(1.5 pt)** Identifique los componentes principales de la aplicación, e indique cómo se relacionan. Para cada componente, indique claramente su propósito. Para especificar cómo se relacionan los componentes, pueden hacer un diagrama informal como el que aparece en la diapo 7 de la clase **11-Componentes.pdf**, o un diagrama de componentes, como el que aparece en la diapo 17.
 - b) **(1.5 pt)** Identifique los patrones de arquitectura usados por la aplicación y haga un diagrama de la arquitectura de la aplicación. Pueden hacer un diagrama informal, como el que aparece en la diapo 10 de la clase **12-Patrones_arq.pdf**. En su diagrama, indique claramente los aspectos de los patrones identificados: por ejemplo, en el diagrama de la diapo 10 se indican las tres capas de la aplicación.
 - c) **(1.0 pt)** Formalice dos requisitos no-funcionales de la aplicación, y explique que estrategia ocuparía para testear que estos requisitos se cumplan. Los dos requisitos deben estar redactados como requisitos formales.
2. Para la segunda parte del control, seleccione un algoritmo canónico de búsqueda u ordenamiento (e.g., quicksort, búsqueda binaria, etc.). Encuentre una implementación de este algoritmo, en el lenguaje de programación que más le acomode². Incluya este código en su respuesta y conteste las siguientes preguntas al respecto:
- a) **(0.5 pt)** Dibuje el control flow graph (CFG) del trozo de código elegido.
 - b) **(0.5 pt)** Especifique un conjunto de casos de prueba que permita alcanzar el 100 % de cobertura para el criterio de *branch*. Deben considerar los posibles caminos escondidos. Si no se puede alcanzar el 100 % de cobertura según este criterio, explique por qué.
 - c) **(1.0 pt)** Ahora especifique un conjunto de casos de prueba que permita alcanzar el 100 % de cobertura para el criterio *all-uses*. Nuevamente, si no se puede alcanzar el 100 % de cobertura, explique por qué.

Instrucciones: deben desarrollar sus respuestas en forma individual, y tienen hasta el **jueves 16 de julio @ 4pm** para entregar. Podemos agendar una sesión de Zoom para responder consultas del enunciado, propongo el martes 13 a las 4:15pm. También pueden hacerme consultas al correo (jsimmond@dcc.uchile.cl) o en el foro del curso. Pueden escribir sus respuestas a mano o tipearlas, lo que les quede más cómodo. Si respondieron a mano, por favor tomar fotos de buena resolución y contraste. En el caso de que hayan tipeado sus respuestas, por favor solo enviar en formato PDF.

¹Los ejemplos son solo para aclarar las categorías, pueden elegir aplicaciones que no aparezcan listadas en el enunciado.

²Busquen una implementación simple, que no tenga más de 20 líneas de código.