

Enunciado Primer Short Paper

Desarrollo de Software

Escuela de Ingeniería Informática - UCAB

Prof. Carlos Alonzo

4 de mayo de 2022

Construir un modelo no es solo elaborar diagramas. El modelo corresponde al conjunto de abstracciones que diseñamos para crear una representación de un problema de la vida real. En dicho modelo representamos objetos que tienen responsabilidades y colaboraciones. El código es el depositario final del diseño o modelo, es la versión codificada del modelo en algún lenguaje de programación.

Índice

1. Lineamientos para la entrega del Short Paper	1
2. Descripción del problema a modelar y codificar en TypeScript	2
3. Descripción del problema a modelar y codificar en Dart	3

1. Lineamientos para la entrega del Short Paper

Para la entrega del primer *Short Paper*, deben seguir las siguientes indicaciones:

- La entrega se hace por subgrupos el día martes 17 de mayo. El paper deben entregarlo impreso y cada subgrupo debe entregar los enlaces a los dos (2) repositorios de GitHub donde se encuentra en código fuente en TypeScript y en Dart. Estos enlaces al Git se deben enviar mediante la bandeja de entrada de M7.

- En esta primera asignación no se trabaja ni con base de datos, ni con interfaces de usuarios ni con acceso a API externas.
- En esta primera asignación no se usa Flutter ni NestJS.
- Solamente se debe usar los lenguajes de programación Dart y TypeScript.
- Obligatoriamente se hace uso del Paradigma Orientado a Objetos.
- Para esta primera asignación se debe hacer uso de los diagramas UML, de todos los que requiera para explicar sus diseños: diagramas de clases, diagramas de secuencia, diagramas de estado.
- **Todos** los integrantes de cada subgrupo **deben tener commits de aportes significativos al código fuente**. Caso contrario, el estudiante tendrá **cero** en la nota de esta actividad.
- En el README de los repositorios Git de los proyectos, deben contruir una tabla donde estén cada uno de los integrantes del subgrupo y se indiquen las tareas que realizaron, destacando los commits más relevantes y detallando el aporte de dichos commits.
- Se evaluará la redacción y ortografía del *Short Paper*.
- Se evaluará el respeto al formato o plantilla que se indica en M7 para el *Short Paper*.
- Se evaluará el orden y la coherencia en la presentación de las ideas y los resultados obtenidos en las diversas iteraciones del diseño.
- Se evaluará la profundidad y validez de las conclusiones presentadas en el *Short Paper*.
- Es **obligatorio** que en el diseño y en el código se hagan uso de los conceptos fundamentales del Diseño Orientado a Objetos: abstracción, encapsulamiento, composición, agregación, delegación, polimorfismo de subtipos, herencia, acoplamiento abstracto, Principios SOLID.
- También se debe hacer uso de la Programación Genérica o Polimorfismo Paramétrico.

2. Descripción del problema a modelar y codificar en TypeScript

A continuación se describe el dominio del problema que vamos a modelar en objetos y codificar en TypeScript. Su objetivo es **identificar objetos (abstracciones) y asignarles responsabilidades y establecer relaciones y colaboraciones entre ellos para cumplir con todos los requerimientos que se describen en este apartado**.

Recuerde que *NO se deben usar base de datos, ni API externas o de terceros, ni usar o definir interfaces de usuario (UI)*. Solo debe construir un modelo de objetos que represente y de vida a los requerimientos que se enumeran en lo siguiente.

- Los Doctores poseen un nombre y una especialidad.
- Los Doctores tienen Pacientes.

- Los Pacientes poseen un nombre, edad, profesión, peso, teléfonos, correo electrónico. Estos datos no los puede modificar ningún Doctor, esto solo lo puede modificar el Paciente.
- Los Doctores manejan citas con los Pacientes. Luego de cada cita el Doctor crea un Registro de Historia Médica del Paciente. Esto no lo puede modificar un Paciente.
- En cada Registro se almacena información diversa, dependiendo de la especialidad.
- Algunos datos de ejemplo que se pueden registrar son: Antecedentes personales, Presión Arterial, Frecuencia Cardíaca, Saturación, Peso, Altura. Algunos de estos datos poseen unidades de medición específicas.
- Según la especialidad se puede registrar valores que puedan ser relevantes, por ejemplo, Colesterol (mg/dL) o Albumina (g/dL). Estos valores pueden tener rangos de referencia.
- Los Doctores solo pueden cambiar los Registro de Historia Médica que ellos hayan creado. Y solo pueden cambiar o crear Registros de sus Pacientes.
- Se debe llevar un registro que permita auditar quién y cuándo ha realizado un cambio en los Registros de Historia Médica de los Pacientes.
- La Historia Médica pertenece al Paciente, y a lo largo del tiempo varios Doctores pueden ir generando Registros de Historia Médica para el Paciente.
- Los Pacientes se pueden subscribir en el sistema de Telemedicina, pero si no están al día con el pago de su suscripción, no pueden hacer uso del sistema, es decir, no pueden agendar citas con especialistas. Los Pacientes pueden cancelar su suscripción al sistema cuando lo deseen.
- El Paciente puede disponer de varios mecanismos de pago y modalidades: Tarjeta de Crédito, PayPal, Pago mensual o anual, Beneficio por paquete de empresa (Beneficio a Empleados), etc.

3. Descripción del problema a modelar y codificar en Dart

A continuación se describe el dominio del problema que vamos a modelar en objetos y codificar en Dart. Su objetivo es **identificar objetos (abstracciones) y asignarles responsabilidades y establecer relaciones y colaboraciones entre ellos para cumplir con todos los requerimientos que se describen en este apartado.**

Recuerde que *NO se deben usar base de datos, ni API externas o de terceros, ni usar o definir interfaces de usuario (UI). Solo debe construir un modelo de objetos que represente y de vida a los requerimientos que se enumeran en lo siguiente.*

- Los Pacientes pueden buscar Doctores por Especialidad y por Ubicación Geográfica (Estado - Ciudad). Si bien el sistema permite realizar citas remotas (Teleconsultas por videollamadas) en algún momento un Paciente puede requerir ir al consultorio.
- Los Pacientes piden citas para Teleconsultas. La suscripción del Paciente debe estar activa y el Paciente no debe tener ningún bloqueo en el sistema (por ejemplo, suspensión por mal uso del servicio).

- Los Pacientes pueden cancelar las citas o pueden no aceptarlas.
- A los Pacientes les llegan notificaciones cuando el Doctor le hace un cambio en su Historia Médica, por ejemplo. Las notificaciones pueden ocurrir para otros eventos en el futuro. Las notificaciones pueden llegar por SMS, Notificaciones Push, Correo, Llamada Automática, etc.
- Dependiendo del evento que se desee notificar, se puede hacer por una o varias vías. Es decir, la cancelación de una cita puede ir por SMS y llamada.
- Los Pacientes poseen un nombre, edad, profesión, peso, teléfonos, correo electrónico. Estos datos no los puede modificar ningún Doctor, esto solo lo puede modificar el Paciente.
- Los Doctores poseen un nombre y una especialidad.
- Toda la actividad de un Paciente se debe registrar para futuros análisis de comportamiento como usuarios del sistema. Por ejemplo, se debe registrar cuáles búsquedas hace por especialidad, cuándo hace una solicitud de cita, cuándo lee su Historia Médica.
- Por cada Teleconsulta, el Paciente pueden hacer un comentario y una valoración del servicio prestado por el Doctor (Especialista) durante la videollamada.
- El Paciente puede buscar el Top 10 de los Doctores mejor evaluados.