



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación
IIC2113: Diseño Detallado de Software

Proyecto Semestral

Objetivos

- Aplicar metodologías ágiles de desarrollo de software con especial énfasis en el diseño de la aplicación.
- Aprender a utilizar el framework ASP.NET Core 2.0 para desarrollar aplicaciones web.
- Diseñar software extensible, mantenible y resiliente a cambios espontáneos en las especificaciones y a cambios del equipo de desarrollo.
- Entender la necesidad del diseño en el desarrollo de una aplicación de software.

Introducción

En el mundo del desarrollo de software y la academia, es común la realización de conferencias que agrupen a autoridades y figuras emblemáticas del área para exponer acerca de las últimas iniciativas e innovaciones en la disciplina a un público especializado interesado. La organización de estas conferencias sin embargo es un proceso extremadamente laborioso, que requiere coordinar esfuerzos de organizadores, expositores, asistentes y los administradores del sitio de la conferencia.

Se le solicita entonces a su equipo de desarrollo desarrollar una aplicación web que permita facilitar el proceso de organizar una conferencia, registrarse a los distintos eventos, y posteriormente analizar los datos de asistencia para propósitos de analytics.

Requisitos Funcionales

La misma aplicación web debe poder ser usada por organizadores, expositores y asistentes para en primer lugar definir los eventos de una conferencia, y posteriormente registrarse como asistente de un evento particular. Se definen tres grupos de funcionalidades: Construcción, Ejecución y Medición.

Funcionalidades de Construcción

En primer lugar, la aplicación debe permitir a los organizadores realizar las siguientes operaciones:

- Crear conferencia: incluye nombre, descripción y patrocinadores. Las conferencias se suelen realizar periódicamente (en general anualmente).
- Crear centro de evento: un centro de evento corresponde al recinto donde se realizará a cabo una instancia de una conferencia. En general se trata de hoteles, universidades, centros de conferencias o recintos similares.
- Crear salas: un centro de evento dispone de múltiples salas para llevar a cabo un evento en particular. Cada sala dispone de una capacidad máxima de asistente y una ubicación, idealmente señalizada mediante un mapa. Adicionalmente, debe definir el tipo de equipamiento disponible para la realización de eventos especializados (proyectores, computadores, enchufes, etc...)
- Crear evento: dada una sala en particular, debe ser posible crear un tipo de evento en algún horario específico. Naturalmente, no pueden realizarse dos eventos en el mismo lugar al mismo tiempo. La aplicación debe soportar varios tipos de eventos:
 - Charlas: consisten en sesiones expositivas donde simplemente se habla de un tema de interés. Se espera que la aplicación permita compartir slides, transcripts, papers u otro material relevante relacionado con la charla a posteriori. Las charlas cuentan con expositores registrados.
 - Sesiones prácticas: son sesiones más largas donde se espera que un especialista guíe a una audiencia en la realización de un ejercicio práctico en tiempo real. Idealmente, todo el material de la sesión (código, tutoriales, links de interés...) debiese hacerse disponible al acceder a la página. Las sesiones prácticas cuentan con expositores y asistentes registrados. Los expositores guían la actividad mientras que los asistentes van respondiendo preguntas y atendiendo dudas particulares.
 - Chats: sesiones más informales y menos estructuradas donde se invita a un equipo de panelistas a debatir y donde se le da la palabra a la audiencia para involucrarse en la conversación. Las sesiones de chat cuentan con panelistas registrados y un moderador.
 - Comidas: todo evento bien organizado debe contar con instancias para compartir con los pares, nada mejor para lo anterior que un almuerzo o cena. Idealmente, la aplicación debe especificar la ubicación y el menú de estos eventos.
 - Fiestas: no todo es trabajo en una conferencia, también puede haber un periodo de relajación informal donde el único propósito sea el pasarlo bien.
- Charlas, sesiones prácticas y chats suelen agruparse por tópico o track. Debe ser posible crear esta organización a través de la aplicación.

- Enviar notificación: debe ser posible enviar notificaciones masivas a grupos de personas en caso de ser necesario comunicar información relevante para la conferencia. Estos grupos pueden ser:
 - Todos los asistentes de la conferencia
 - Todos los asistentes registrados para un evento en particular
 - Todos los expositores
 - Todos los expositores de un evento en particular
- Crear página de orador: para todos los participantes en una conferencia debe ser posible crear una página donde se presente una breve biografía del expositor, currículum e información de contacto.
- Crear foro: se espera que la aplicación entregue un espacio para que los asistentes puedan compartir comentarios y dar retroalimentación sobre su experiencia.
- Definir hospedaje: muchos centros de eventos son o tienen asociación con un hotel que ofrece hospedaje para los asistentes. En caso de ofrecer servicio de hospedaje, un organizador debe especificar qué tipos de habitaciones hay disponible, las características de cada una de ellas, sus respectivas disponibilidades y el costo.

Funcionalidades de Ejecución

Una vez finalizada la organización de una conferencia, los asistentes deben poder acceder a la aplicación para realizar las siguientes actividades:

- Registrarse para atender un evento en particular. Naturalmente, no debe ser posible registrarse a dos eventos distintos en la misma hora. Si se trata de una comida, se debe poder elegir una opción del menú.
- Revisar mi itinerario completo de forma cómoda.
- Participar en foros.
- Entregar retroalimentación de la conferencia o un evento en particular. La retroalimentación para los eventos debe admitir comentarios generales y una nota que el asistente puede entregar en distintos ámbitos (e.g. calidad de la conferencia, calidad del expositor, comodidad del recinto, calidad de la comida...)
- En caso de existir opción de hospedaje, el asistente debe ser capaz de reservar una habitación. Se puede omitir para este proyecto la creación de una instancia de pago.

Funcionalidades de Medición

Por último, organizadores deben poder acceder a herramientas de analytics de la aplicación para determinar:

- Métricas de asistencia a los distintos tipos de eventos y tracks de la conferencia
- Métricas de calidad de los distintos tipos de eventos de la conferencia y tracks según calificaciones ingresadas por los asistentes
- Métricas de calidad de los distintos expositores
- **Cómo han ido evolucionando las distintas métricas anteriores entre instancias distintas de la misma conferencia**
- Esta herramienta debe ser compatible con que cualquier dato o label pueda ser agregado, editado o borrado en cualquier momento

Toda información ingresada por un administrador u organizador debe admitir poder ser editada o borrada en cualquier momento. Por ejemplo, no es aceptable impedir el modificar el horario de un evento si alguien ya se registró para atender el evento. Las implicancias de lo anterior en el diseño de la solución deben ser contempladas por los alumnos. Hay libertad para manejar estos casos, siempre y cuando su propuesta sea consistente y se minimicen los efectos colaterales en los usuarios.

Requisitos Mínimos de Desarrollo

Durante el desarrollo de este proyecto, se le pide tomar en cuenta las siguientes restricciones en la elaboración de su solución:

- ASP.NET Core 2.0

En el desarrollo de su solución, usted deberá utilizar la tecnología ASP.NET Core 2.0 en su modalidad MVC o con Razor Pages. Se deja la opción de usar o no un framework de Frontend también como React, Angular o Vue.

- Kanban: Trello

Utilizar el servicio de Kanban Trello para organizar su trabajo como equipo. Cada equipo debe tener un tablero de Trello que compartirá con su product owner. Este puede tener la estructura (columnas) que el equipo encuentre conveniente mientras se note un claro flujo de trabajo que comunique el estado del proyecto a su product owner.

- Gitflow

Para desarrollar la aplicación, gestionarán su proyecto en un repositorio git. Sobre esto, deben seguir el modelo de uso de Gitflow. No es necesario seguirlo al pie de la letra, mientras se ocupen al menos dos branches principales una branch por funcionalidad.

Bonus

Se ofrece un bonus de 0.5 puntos en la nota de presentación de la entrega 4 a los grupos que presenten su aplicación corriendo sobre Microsoft Azure.

Entregas y Evaluación

1. Entrega 1 (10%)

Mockups del flujo completo de su aplicación utilizando una herramienta digital de wireframing (e.g. Balsamiq Mockups).

2. Entrega 2 (20%)

Diseño

El grupo deberá explicarle a su Product Owner, con el apoyo de un diagrama de diseño completo, más los diagramas UML que estimen convenientes, cómo es que el diseño de su aplicación permite cumplir con todas las funcionalidades de la sección “Funcionalidades de Medición” del enunciado. Se evaluará correctitud y completitud del diseño, además del no improvisar aspectos del diseño durante la evaluación.

[50% de la nota de la entrega]

Producto

Además, deberán mostrar una primera versión de su aplicación funcionando en localhost. Esta debe contener el landing page de su página, funcionalidad de registro y login, el sistema de navegación a los distintos puntos de su página y el modelo ya implementados.

[50% de la nota de la entrega]

3. Entrega 3 (30%)

Diseño

El newcomer del grupo deberá explicarle a su Product Owner, con el apoyo de un diagrama de diseño completo más los diagramas UML que estimen convenientes, cómo es que el diseño de su aplicación permite cumplir con todas las funcionalidades de la sección “Funcionalidades de Medición” del enunciado. Se evaluará correctitud y completitud del diseño, además del no improvisar aspectos del diseño durante la evaluación. Se evaluará también el haber corregido todas las deficiencias del diseño propuesto en la entrega 2. Durante la evaluación, queda estrictamente prohibido que algún miembro del grupo distinto al newcomer intervenga.

[30% de la nota de la entrega]

Producto

Segunda versión funcionando en localhost. Esta debe contener el 50% de las funcionalidades aquí descritas implementadas. Negociar específicamente cuáles con Product Owner.

[40% de la nota de la entrega]

Presentación

Adicionalmente, se deberá realizar una presentación de 6 minutos que incluya principalmente una demo práctica de su aplicación en formato de “pitch” orientado a potenciales inversionistas.

[30% de la nota de la entrega]

4. Entrega 4 (40%)

Diseño

El newcomer del grupo deberá explicarle a su Product Owner, con el apoyo de un diagrama de diseño completo más los diagramas UML que estimen convenientes, cómo es que el diseño de su aplicación permite cumplir con todas las funcionalidades de la sección “Funcionalidades de Medición” del enunciado. Se evaluará correctitud y completitud del diseño, además del no improvisar aspectos del diseño durante la evaluación. Se evaluará también el haber corregido todas las deficiencias del diseño propuesto en la entrega 3. Durante la evaluación, queda estrictamente prohibido que algún miembro del grupo distinto al newcomer intervenga.

[20% de la nota de la entrega]

Producto

Última versión de su aplicación funcionando en localhost. Esta debe contener el 100% de las funcionalidades aquí descritas implementadas.

[40% de la nota de la entrega]

Presentación

Adicionalmente, se deberá realizar una presentación de 10 minutos que incluya principalmente una demo práctica de su aplicación en formato de “pitch” orientado a potenciales inversionistas. Grupos que tengan su aplicación corriendo en Azure durante la demo recibirán un bonus en esta evaluación.

[40% de la nota de la entrega]

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería en el SIDING.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1.1. en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona. Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente. Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.