

Programa IIC2513 Tecnologías y Aplicaciones Web

Primer semestre 2021

Curso

Nombre: IIC2513 - Tecnologías y Aplicaciones Web

Sección: 2

Horario: M-J: 3

Créditos: 10

Requisitos: IIC2412 o IIC2413

Equipo docente

Profesor cátedra: Hernán Cabrera - hernan.cabrera@uc.cl

Ayudante coordinación: Valeria Riquelme

Descripción del curso

En la actualidad vivimos en un mundo hiperconectado gracias a Internet. Es muy común que, por temas de estudios, trabajo, entretenimiento, trámites u otros, accedemos a distintos sitios web en búsqueda de información e interacción.

Estos sitios suelen tener distintos objetivos: compartir publicaciones y/o imágenes, vender productos, realizar transacciones, proveer comunicación en tiempo real, entre otros. A la vez, son diferentes en cuanto a comportamiento: en algunos casos hacer clic en un enlace gatilla una recarga del sitio, y en otros puede desplegar información inmediatamente. Además de lo anterior, puede que incluso se vean diferente según el dispositivo que estemos utilizando (computador, teléfono, tablet, etc). Lo anterior es sólo una muestra de la gran diversidad presente en los sitios web. Este hecho se relaciona con las distintas tecnologías que están involucradas en la World Wide Web (WWW).

La construcción de estos sitios o, en muchos casos, aplicaciones, requiere entender cómo funciona la Web y cuáles son las tecnologías disponibles para su desarrollo, de forma de poder utilizarlas en la construcción de software seguro y de calidad sobre la Web. Este curso aborda principalmente estos temas.

A lo largo del curso se dará atención a la estructura cliente-servidor del entorno web, detallando lenguajes de programación y frameworks tanto para front-end como back-end, junto con explicaciones detalladas de los protocolos involucrados. El curso también aborda el diseño y construcción de APIs que permitan el intercambio de información entre un cliente y un servidor. Se dará también énfasis a la construcción de aplicaciones web pensando en el usuario final (User Experience o UX) y al proceso y ciclo de desarrollo software, incorporando conceptos de calidad con pruebas unitarias, de integración y de sistemas. Además, se abordará la construcción y codificación de aplicaciones web seguras capaces de hacer frente a los ataques maliciosos más comunes.

Objetivo

El objetivo de este curso es que los estudiantes conozcan la infraestructura tecnológica sobre la cual descansa la World Wide Web y aprendan a manejar las principales tecnologías que se utilizan para construir sitios y aplicaciones seguras y de calidad en la plataforma web. Además, adquirirán una base conceptual suficiente como para que puedan entender las nuevas tecnologías y propuestas que surgen día a día.

Resultados del aprendizaje

Al finalizar el curso el estudiante será capaz de:

- Entender la arquitectura tecnológica de las páginas y aplicaciones web.
- Conocer los principales estándares que hacen posible la WWW.
- Entender las tecnologías, herramientas, frameworks y métodos involucrados en el desarrollo del front-end y back-end de una aplicación web moderna.
- Conocer los protocolos que rigen las aplicaciones WEB.
- Entender, diseñar y utilizar recursos de la Web a través de APIs ya sean de construcción propia como entregadas por terceros.
- Desarrollar sitios web con contenido dinámico y aplicaciones web utilizando diversas tecnologías y herramientas modernas, disponibles en la actualidad, para este fin.
- Entender los estándares y modelos para la verificación y validación de una aplicación web.
- Proponer y aplicar un plan de verificación y validación para un proyecto de desarrollo de una página o aplicación web.
- Identificar vulnerabilidades en las aplicaciones web y escribir código para defenderlas de los ataques.
- Evaluar los méritos de las nuevas tecnologías para el desarrollo de sitios y aplicaciones web, como también aquellas que soportan la WWW.

Contenidos

1. Fundamentos de la WWW
 - Arquitectura de una aplicación Web típica (modelo MVC)
 - Lenguajes y *frameworks* para desarrollo Web
 - Protocolos HTTP, TCP, IP
 - HTML y CSS
2. Introducción a la plataforma NodeJS y a *middlewares* Koa y Express
 - El lenguaje JavaScript
 - JavaScript en el lado del servidor: NodeJS
 - Middleware* en el lado del servidor: Koa y Express
3. Desarrollo de una aplicación con NodeJS y Koa
 - Controladores y Vistas
 - Modelo de datos y ORM
 - Clases, tablas, migraciones
 - Incorporación de un front end básico (HTML, CSS, JavaScript)

4. Aseguramiento de la Calidad
 - Conceptos básicos Calidad de software
 - Revisiones de código (en general)
 - Revisión de código con Github (uso de pull requests)
 - Pruebas unitarias con Jest/Mocha/Cypress
 - Pruebas de integración y pruebas a nivel de sistema
5. Construyendo servicios Web y APIs
 - La importancia de las APIs
 - El modelo REST
 - Diseño de una API *RESTful*
 - GraphQL, Query language para APIs
6. Aplicaciones web de cliente enriquecido (RIA)
 - JavaScript y el DOM
 - Frameworks* del lado del cliente
 - React, biblioteca (library) para construir aplicaciones
 - VueJs, framework para front-end
7. Seguridad
 - La importancia de construir software seguro
 - Principales vulnerabilidades de las aplicaciones web
 - Protocolos seguros en la Web

Metodología

A lo largo del semestre el profesor presentará los contenidos del curso, tanto de forma teórica como de forma práctica, por medio de cátedras. Además, existirán cápsulas complementarias, que abordarán parte importante de los contenidos del curso. **El contenido de estas cápsulas es parte integral del material docente y por lo tanto no se consideran opcionales.** A la vez, los estudiantes desarrollarán actividades y ejercicios propuestos para reforzar el aprendizaje.

Por otra parte, trabajarán en equipos de 3 personas, desarrollando una aplicación web propia. Las distintas tecnologías, estándares y técnicas van siendo introducidas a medida que las aplicaciones se completan, en el momento que se hacen necesarias. En forma periódica y previamente informada, los estudiantes serán evaluados por su grado de avance en los proyectos.

Los estudiantes deberán llevar a cabo un **trabajo personal significativo**, (complementario a las clases) que involucra:

- Lectura y estudio de aspectos específicos del lenguaje, del framework y/o de las diferentes tecnologías involucradas.
- Lecturas y actividades complementarias que profundizan y amplían lo que se discute en clases.
- Desarrollo de proyectos a lo largo del semestre.

Proyecto

A grandes rasgos, el proyecto considerará:

- Construir una aplicación web sencilla con un front-end básico (HTML, CSS y JS lado cliente) y con NodeJS y Koa que entregarán páginas WEB estáticas.

- Ampliar y construir una aplicación web completa que considere acceso a base de datos (ORM), rutas a recursos, autenticación (login usuario), seguridad, encriptación, APIs y un lado cliente construido principalmente en React, incorporando aspectos de calidad y seguridad

Evaluación

Los estudiantes serán evaluados en dos dimensiones: **teórica y práctica**.

Para la parte teórica se llevará a cabo una única interrogación, en calendario de evaluaciones corresponde a la I2, que cubre el material visto en clases, cápsulas y lecturas, hasta la semana anterior a la fecha de la interrogación.

La interrogación no será presencia y constará de dos partes, una parte teórica (preguntas de contenido) y una parte práctica que corresponderá a un trabajo individual de desarrollo. La interrogación se evaluará con nota de 1 a 7 usando una escala continua. El día de la interrogación no habrá clases del curso.

Para la parte práctica cada entrega de proyecto será evaluada con una nota de 1 a 7 siguiendo una escala discreta que se detalla a continuación:

Calificación	Nota asociada	Observaciones
A	7,0	La entrega excede con creces lo solicitado.
B+	6,0	La entrega cumple algo más allá de lo solicitado
B	5,5	La entrega cumple con lo mínimo solicitado en el enunciado.
C	4,0	La entrega cumple parcialmente con lo mínimo solicitado.
D	2,5	La entrega está muy por debajo del mínimo solicitado.
E	1,0	No se realizó la entrega en los plazos.

Las calificaciones del trabajo práctico y los cumplimientos asociados a cada calificación serán **guiados activamente por el cuerpo docente, principalmente a través de los/las ayudantes**.

Posterior a la calificación de cada entrega grupal, los alumnos integrantes del grupo deberán calificarse entre sí en una **evaluación de pares** que medirá diversos aspectos de cumplimiento de trabajo en grupo. La evaluación de pares puede mantener (de forma individual) la nota conseguida por el grupo o disminuirla de manera continua. La evaluación de pares varía entre 0 y 2,0. Si alguien obtiene 0 en su evaluación de pares, la nota asociada para esa entrega, para esa persona, será de 1,0.

La nota de evaluación de pares asociada a una entrega de una persona corresponderá al promedio de la calificación de pares obtenida por esta persona en esa entrega. Así, si a una persona la evalúan con 1,8 y 1,7, su evaluación de pares será de 1,75.

Si uno o más estudiantes no entrega la evaluación de pares correspondiente a una entrega de su grupo, se les corregirá la entrega, pero no se le informará la calificación al grupo hasta que todas las evaluaciones de pares hayan sido realizadas.

El cálculo de la nota individual, luego de la evaluación de pares es:

$$[(\text{Nota_Grupal} - 1) * \text{Evaluacion_de_Pares}] / 2 + 1$$

Por ejemplo, si un grupo obtuvo un 7 en la entrega, y una persona del grupo recibió un 1,75 en la evaluación de pares, su nota será $(7-1)*1,75/2 + 1 = 6,25$ (es decir, un 6,3)

Las entregas parciales del proyecto (luego de evaluación de pares) serán promediadas y eso generará la calificación del proyecto en particular, llamado nota práctica.

La nota práctica (NP) deberá ser **igual o superior a 4,0**. Una nota inferior a 4,0 significa la reprobación del curso.

El examen final es una evaluación que cubre todo el material del semestre. Al igual que la interrogación tendrá una parte teórica (preguntas de contenido) y una parte práctica que corresponderá a un trabajo individual de desarrollo. Adicionalmente, la nota del examen puede reemplazar la nota de la interrogación, en caso de ser esta última menor o de no haberse rendido. **En caso de inasistencia al examen, debe ser justificada ante la Escuela de Ingeniería según el procedimiento oficial.**

La nota final considera la interrogación, el examen y el proyecto, de la siguiente forma:

- **Nota Teórica (NT):** $(\text{Interrogación} + 2 * \text{Ex} - \min(\text{Interrogación}, \text{Ex})) / 2$
- **Nota Práctica (NP)**
- **Nota final (NF):**
 - Si notas Teórica y Práctica son mayores o iguales a 4,0: **$\text{NF} = \text{NP} * 0.6 + \text{NT} * 0.4$**
 - En caso contrario, la menor entre ambas

Durante el semestre podría haber evaluaciones y/o actividades que, aunque no serán consideradas directamente en la nota final, podrían significar bonificaciones en alguna de las evaluaciones descritas anteriormente.

Fechas de evaluaciones

- **Interrogación:** jueves 26 de mayo
- **Examen final:** martes 05 de julio

Fechas entregas de proyecto

- **Entrega 0:** jueves 31 de marzo (diseño)
- **Entrega 1:** miércoles 20 de abril
- **Entrega 2:** jueves 12 de mayo
- **Entrega 3:** jueves 02 de junio
- **Entrega 4:** viernes 17 de junio

Política de atrasos

Los estudiantes tendrán la posibilidad de entregar sus trabajos con atraso, con modalidades diferentes para evaluaciones y proyectos:

- **Interrogación y examen:** el primer día de atraso **descontará un 25% del puntaje total obtenido**. Un día de retraso comienza al segundo siguiente del plazo establecido (ejemplo: si el plazo es a las 22:00:00, entonces entregar a las 22:00:01 ya se considera un día de atraso). Al segundo día de atraso, se califica la evaluación con nota 1.0.
 - **Proyecto:** cada grupo dispondrá de **4 cupones de atraso** durante el semestre, donde cada cupón le permitirá acceder a un día **HÁBIL** de atraso (24 horas estrictas). Un día de atraso comienza de la misma forma que para las evaluaciones. En otras palabras, cada grupo tendrá derecho a 4 días de atraso en total durante el semestre, que puede distribuir en las distintas entregas de los proyectos de la manera que estime más conveniente.
- Nota:** Para la última entrega se permitirá el uso de un máximo de dos cupones de atraso.

Código de Honor

Este curso se adscribe al Código de Honor firmado por la Escuela de Ingeniería el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso de que exista colaboración permitida con otros estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es su deber conocer la [versión en línea del Código de Honor](#). Puede encontrar más información en la plataforma Siding (Pregrado - Integridad Académica - Integridad Académica).

Bibliografía

No hay bibliografía mínima. Toda la bibliografía es complementaria.

- Haverbeke M. "Eloquent JavaScript, 3rd Ed.: A Modern Introduction to Programming" No Starch Press 2018
- Crockford D. "JavaScript: The Good Parts" O'Reilly 2008
- Flanagan D. "JavaScript: The Definitive Guide: Activate Your Web Pages" O'Reilly 2011
- Simpson, K. "You Don't Know JS" O'Reilly 2015
- O. Omole "Server Side development with Node.js and Koa.js", Packt Publishing 2018
- Cantelon M., Meck B., Young A. "Node.js in Action" Second Edition Manning 2017
- Lawson B., Sharp R. "Introducing HTML5", New Riders 2011
- Keith J. "HTML 5 for Web Designers" A book apart 2011
- Dan Mantyla, "Functional Programming in JavaScript", Packt Publishing, Limited 2015.
- Powers, S. "Learning JavaScript" O'Reilly 2006
- Adam Freeman, "Pro React 16", Apress 2019

- Lawrence D. Spencer and Seth H. Richards, "Reliable JavaScript : How to Code Safely in the World's Most Dangerous Language", John Wiley & Sons, Incorporated 2015.
- Cory. Gackenheime, "Introduction to React", Apress 2015
- A. Hoffman, "Web Application Security: Exploitation and Countermeasures for Modern Web Applications", O'Reilly 2020
- C. Laporte, A. April, "Software Quality Assurance", Wiley IEEE Press 2018