

| | |
|--|--|
| | Cenfotec Diplomado en diseño y desarrollo Web |
|--|--|

WEB03 Fundamentos de programación web

Periodo: Primer cuatrimestre

Modalidad: Cuatrimestral

Requisitos: Admisión a la carrera

Naturaleza del curso: teórico-práctico

Créditos: 5

Horas de clase por semana: 6 horas

Horas de trabajo individual por semana: 9 horas

Asistencia: obligatoria

Puede presentarse por suficiencia: sí

Se permite hacer examen extraordinario: sí

Vigencia: Enero del 2013-presente

Profesores: (se especificará cada cuatrimestre)

Laboratorio: (se especificará cada cuatrimestre)

Horario: (se especificará cada cuatrimestre)

Descripción del curso

Este curso prepara y entrena al estudiante en la construcción de programas de software para la Web, expresados en los paradigmas de programación imperativo y estructurado, usando el lenguaje de programación JavaScript como herramienta de construcción de soluciones a problemas pequeños y medianos, a fin de establecer las bases para resolver problemas más complejos mediante programación.

Objetivo general

Encontrar soluciones a problemas de programación pequeños y medianos, que involucren validar formularios y hacer procesamiento básico para la interfaz de un sitio Web, al elaborar programas de software expresados en estructuras secuenciales, estructuras condicionales, estructuras iterativas, funciones, hileras y arreglos, usando el lenguaje de programación JavaScript.

Objetivos específicos

- Aplicar los fundamentos de la solución sistemática de problemas pequeños por medios algorítmicos.
- Desarrollar programas utilizando el paradigma de programación estructurada, usando el lenguaje de programación JavaScript.
- Utilizar buenas prácticas en la elaboración de programas tales como el uso de estándares,

buen estilo de codificación y técnicas básicas para el aseguramiento de la calidad.
Apreciar la claridad, estructura, legibilidad, eficiencia y simplicidad de los algoritmos y programas de software, elaborados bajo el paradigma de programación estructurada.
Criticar constructivamente las soluciones, algoritmos y programas propuestos por sus compañeros, aplicando un proceso de análisis y revisión conjunta de algoritmos y programas.

Contenidos

Introducción a la programación del lado del cliente
Conceptos básicos de programación.
Valores, operadores y expresiones.
Variables, constantes y asignación.
Conceptos de algoritmo y de programa.
Ejecución de un programa.
Estructuras de programación básicas.
Estructuras secuenciales.
Estructuras condicionales.
Estructuras iterativas.
Uso de las estructuras de programación básicas.
Resolución algorítmica de problemas.
Programación en JavaScript.
Sintaxis básica.
Valores y tipos, operadores y expresiones.
Ejecución de programas JavaScript (ECMAScript) y uso de JavaScript Shell.
Convenciones y estándares de codificación.
Uso de JSLint.
Estructuras de control básicas.
Estructuras secuenciales.
Estructuras condicionales.
Estructuras iterativas.
Uso de las estructuras de control.
Abstracción procedimental: funciones y parámetros en JavaScript.
Arreglos y matrices.
Problemas con arreglos y funciones.
Documentación interna del programa.
Introducción a las expresiones regulares en JavaScript.
Introducción al Document Object Model.
Validación de formularios HTML.
Control de calidad.
Listas de revisión.
Introducción a las pruebas de unidad.
Depuración de código.
Uso de herramientas para ejecución, depuración, prueba y análisis de JavaScript:
JavaScript Shell, Firebug, YSlow, Web Developer, Aptana Studio.

Metodología

El trabajo en clase se distribuye en exposiciones magistrales por parte del profesor seguidas por consultas a los estudiantes en las que el profesor tiene la oportunidad de aclarar dudas y fomentar la participación de los estudiantes. El profesor también compara y

contrasta diferentes ejemplos en cada tema para ayudar a aclarar los conocimientos y facilitar su comprensión.

Las sesiones magistrales son frecuentemente seguidas por prácticas individuales en clase, en donde los estudiantes pueden reafirmar el conocimiento aplicándolo en las prácticas.

Adicionalmente, en clase se hacen laboratorios en computadora, que giran alrededor de asignaciones fijadas por el profesor, diseñadas para reforzar los conocimientos y aplicarlos. Durante los laboratorios, los estudiantes trabajan bajo la supervisión del profesor, teniéndolo a su disposición para ser consultado cuando se requiera. En los laboratorios, los estudiantes deben resolver problemas pequeños o medianos, y elaborar programas de software usando el paradigma de programación estructurada y el lenguaje de programación JavaScript, en un ambiente controlado. Después de cada laboratorio se hace un análisis de maneras apropiadas de solucionar el problema, para que los estudiantes puedan corregir, completar o verificar las soluciones que ellos han planteado.

El trabajo extra-clase individual y en grupo, puede generarse por asignaciones establecidas por el profesor o por segregación del trabajo en clase. Este trabajo extra-clase corresponde a tareas individuales y tareas en grupo que consisten en elaborar soluciones a problemas pequeños y medianos usando el paradigma de programación estructurada y el lenguaje de programación JavaScript.

El profesor fomentará en los estudiantes, tanto en clase como en los trabajos extra-clase, el análisis y la valoración de la calidad de los algoritmos y programas, usando protocolos de revisión que favorecen la interacción constructiva y la colaboración entre estudiantes en busca de la calidad en su trabajo.

Los problemas que los estudiantes tendrán que resolver en los laboratorios en clase, así como en las tareas individuales y en grupo, serán sencillos al principio e irán incrementando su dificultad, hasta que los estudiantes deban resolver problemas medianos en tamaño y complejidad.

Estrategias de aprendizaje

Las estrategias de aprendizaje que el estudiante debe adoptar para lograr cumplir los objetivos del curso son:

Prácticas en clase:

Ejercicios prácticos que se realizan en clase de manera individual, en donde los estudiantes deben aplicar el conocimiento aprendido.

Laboratorios en clase:

Son laboratorios desarrollados por los estudiantes en las computadoras de una clase equipada, y bajo la guía y supervisión del profesor, estos miden de una manera más formal que las prácticas en clase el conocimiento adquirido por el estudiante.

Tareas individuales y en grupo extra-clase:

Son asignaciones establecidas por el profesor, en las cuales el estudiante o el grupo tiene que elaborar una solución a problemas pequeños y medianos usando el paradigma de programación estructurada y el lenguaje de programación JavaScript.

Lecturas extra-clase:

Las lecturas extra-clase de los libros de texto propuestos en la bibliografía y de sitios de Internet que el profesor considere pertinentes, le ayudarán a los estudiantes a reforzar lo visto en clase. Algunas de las lecturas son en inglés.

Elaboración de proyecto

La elaboración del proyecto requiere que los estudiantes trabajen en grupos pequeños con un proceso en que primero se analizan el problema a solucionar, y luego se formula el desarrollo del mismo, logrando una aplicación web que cuente con funcionalidad realizada en lenguaje de programación.

Recursos didácticos

Se utilizan, para todas las sesiones del curso, presentaciones en proyector digital, que contienen los conceptos correspondientes a cada sesión.

Se utiliza un laboratorio de computadoras debidamente instalado y configurado para realizar los laboratorios en clase, con un ambiente de software que permite ejecutar algoritmos expresados en diagramas de flujo y con otro ambiente para programar en el lenguaje JavaScript.

Se usan recursos que están disponibles en Internet, tales como estándares de codificación, plantillas para las listas de revisión, herramientas para ejecutar programas y casos de prueba.

Se hará uso de los libros, físicos y digitales, que están en la biblioteca para consulta de los estudiantes.

Evaluación

A continuación se especifica la evaluación del curso, teniendo en cuenta su función sumativa, diagnóstica y formativa:

| Concepto | Ponderación |
|--------------|-------------|
| Quices | 10% |
| Laboratorios | 20% |
| Tareas | 10% |
| Examen 1 | 15% |
| Examen 2 | 15% |
| Proyecto | 30% |
| Total | 100% |

Las prácticas en clase se realizan semanalmente y son individuales. Todas las prácticas son supervisadas por el profesor, pero unas son corregidas por los propios estudiantes en clase, comparando la solución presentada por el profesor con la de ellos; otras son corregidas con la colaboración de todos los estudiantes, pues construyen la solución por medio de la participación de todos; y otras son entregadas al profesor para ser calificadas. Estas prácticas le sirven al profesor para diagnosticar si los estudiantes aplican bien los conocimientos aprendidos y, si es necesario, poder recomendar a los estudiantes prácticas adicionales. En estas prácticas los estudiantes tienen la oportunidad de apreciar las soluciones planteadas por los compañeros y por el profesor, para valorar y aplicar aspectos de la calidad de la programación aprendidos. El profesor debe llevar un control de la participación de los estudiantes en las prácticas en clase.

Los laboratorios en clase son desarrollados por los estudiantes en computadora, de manera individual o en grupo, y bajo la guía y supervisión del profesor. En los laboratorios los estudiantes deben resolver algorítmicamente un problema pequeño o mediano y desarrollar un programa en computadora que implemente la solución. Durante cada laboratorio se aclaran las dudas que los estudiantes puedan tener y después de cada laboratorio el profesor, con ayuda de los estudiantes, verifica la solución correcta al problema planteado. Los estudiantes deben entregar los algoritmos y programas elaborados en el laboratorio al profesor, quien los califica. Se hacen 4 laboratorios en total

en el curso, 2 de ellos se hacen de manera individual y los 2 restantes se hacen en grupo. Las tareas individuales y en grupo consisten en elaborar una solución a problemas pequeños y medianos usando el paradigma de programación estructurada y el lenguaje de programación JavaScript. Las tareas son desarrolladas extra-clase, y los estudiantes deben entregar los algoritmos, la documentación y programas elaborados al resolver la tarea. Se hacen 7 tareas en total en el curso, 2 de ellas se hacen de manera individual y las 5 restantes se hacen en grupo. La última tarea es un Mini-proyecto que evalúa todo el conocimiento que se espera comprenda el estudiante; el Mini-proyecto será desarrollado en grupo.

Los exámenes evalúan principalmente los temas que los preceden, pero se pueden considerar que son acumulativos. Los exámenes son individuales, se presentan por escrito y están especificados en el cronograma del curso – pero serán realizados en sesiones extraordinarias (fuera del horario lectivo).

El curso se aprueba con una nota ponderada mayor o igual a 70 (≥ 70). El estudiante que obtenga una nota ponderada mayor o igual a sesenta (60) y menor a setenta (70), tendrá derecho a un examen extraordinario, según se estipula en el Reglamento Académico. Además, se debe respetar el Reglamento del Régimen Estudiantil vigente.

Cronograma

| Semana | Contenidos temáticos | Actividades de aprendizaje |
|-----------|--|---|
| Semana 1 | Introducción a JavaScript Conceptos básicos de programación Valores, operadores y expresiones. Variables, constantes y asignación. | |
| Semana 2 | Estructuras de programación básicas: if-else (algorítmicamente) if-else (codificación) | Quiz 1: fundamentos de programación Tarea 1 : Resolución de problemas |
| Semana 3 | Resolución algorítmica de problemas por medio de la estructura if-else Switch-case (algorítmicamente) | Revisión y solución de la tarea 1 Quiz 2: ejercicio de codificación if-else Tarea2: Resolución de problemas |
| Semana 4 | Switch-case (codificación) Funciones y parámetros en JavaScript | Revisión y solución de la tarea 2 Laboratorio 1: if-else y switch case |
| Semana 5 | Funciones y parámetros | Examen 1: Resolución algorítmica de problemas |
| Semana 6 | Arreglos | Quiz 3: funciones y parámetros |
| Semana 7 | Arreglos | Laboratorio 2 |
| Semana 8 | Matrices | |
| Semana 9 | Matrices Problemas con arreglos, matrices y funciones | Laboratorio 3: Arreglos |
| Semana 10 | Introducción al Document Object Model Métodos de manipulación del DOM | Examen 2 |
| Semana 11 | Introducción a las expresiones regulares en JavaScript | |
| Semana 12 | Validación de formularios HTML | Laboratorio 3: Expresiones |

| | | |
|-----------|--------------------------------|---|
| | | regulares |
| Semana 13 | Validación de formularios HTML | |
| Semana 14 | Control de calidad | Laboratorio 4: Consolidación de conceptos básicos de programación Web en JavaScript |
| Semana 15 | Prácticas de programación Web | Examen 3 Revisión del proyecto |

Bibliografía

Referencias principales:

Morrison, M. (2008). *Head First JavaScript*. Sebastopol, Estados Unidos: O'Reilly Media.

Yank, K., & Adams, C. (2007). *Simply JavaScript*. Australia: SitePoint.

Referencias complementarias:

Code Conventions for the JavaScript Programming Language. (Sin fecha). Consultado el 31 de agosto del 2010 del sitio web de Crockford, <http://javascript.crockford.com/code.html>

Curriculum. (Sin fecha). Consultado el 31 de agosto del 2010 del sitio web de WaSP InterAct, HYPERLINK "<http://interact.webstandards.org/curriculum/front-end-development/dom-scripting-1>" <http://interact.webstandards.org/curriculum/front-end-development/dom-scripting-1>

Drupal handbook pages. JavaScript coding standards. Consultado el 02.07.2010 del sitio de Drupal, HYPERLINK "<http://drupal.org/node/172169>" <http://drupal.org/node/172169>

