

Plan de asignatura PROGRAMACIÓN DE ESTRUCTURAS DINÁMICAS

Departamento de Electrónica e Informática - Modalidad en línea

I. Datos generales de la asignatura

La asignatura Programación de Estructuras Dinámicas pertenece al Departamento de Electrónica e Informática. Dada la situación causada por la pandemia y en atención a las recomendaciones del Ministerio de Educación, Ciencia y Tecnología, las clases del ciclo 02/2021 se desarrollarán de manera virtual/semipresencial durante todo el ciclo.

Unidades Valorativas y horas de dedicación semanal.	La asignatura Programación de Estructuras Dinámicas es de 4 U.V. Para el ciclo 02-2021, en modalidad virtual, exige una dedicación promedio de 8 horas semanales.
Horarios y sesiones síncronas	Secciones 01 y 02. Se realizarán dos sesiones síncronas por semana con los profesores. Se llevarán a cabo los días martes y jueves, a las 3:30 pm. Cada sesión síncrona tendrá una duración aproximada de dos horas clase (100 minutos). La herramienta a utilizar para dichas sesiones es Zoom .
Plataformas de la asignatura	Material de la asignatura y exámenes en línea: https://ecampus.uca.edu.sv/
Docentes de la asignatura.	Ronaldo Armando Canizales Turcios - rcanizales@uca.edu.sv Guillermo Ernesto Cortes Villeda - gcortes@uca.edu.sv
Instructores de la asignatura.	Ricardo Alberto Gómez Quijano, 00108517@uca.edu.sv Oscar Alejandro Rodríguez Abrego, 00206019@uca.edu.sv Diana Sugelyth Umaña Rodríguez, 00143619@uca.edu.sv Ivette Carolina Pinto León, 00186319@uca.edu.sv Fernando Miguel Rodríguez González 00211919@uca.edu.sv Daniel Alejandro Orozco Orellana 00200617@uca.edu.sv
Instancias superiores.	En caso durante el ciclo 02/2020 se presenten situaciones en la asignatura que el docente no pueda resolver puede remitir su consulta al jefe de departamento en primera instancia o al decanato de ingeniería y arquitectura en segunda instancia. <ul style="list-style-type: none">• Jefe de Departamento de Electrónica e Informática: Carlos Anibal Juárez – cjuarez@uca.edu.sv• Decano de la Facultad de Ingeniería y Arquitectura: Carlos Cañas - decanato.ingenieriayarquitectura@uca.edu.sv

II. Presentación de la asignatura

Esta es la segunda de las asignaturas de la rama de programación de computadoras. Es importante en este nivel porque con ella se continúa el desarrollo del pensamiento lógico que necesitan para aprender a programar y se conocen ciertas estructuras y técnicas fundamentales para el manejo de la memoria dinámica. Los materiales principales del curso se han colocado en documentos pdf, pero se acompañan con videos explicativos y enlaces a sitios de Internet que ejemplifican y refuerzan los conceptos y técnicas que se deben aprender. El objetivo del curso es que el estudiante conozca las estructuras de datos fundamentales y que sepa implementarlas por medio de programas de computadora que resuelvan problemas aplicando técnicas modernas de programación. Para esta modalidad virtual mantendremos dichos objetivos pues son la base de otras materias que cursará a lo largo de sus estudios universitarios. Le sacaremos provecho a esta modalidad virtual haciendo uso de los encuentros síncronos donde desarrollaremos juntos los temas de interés y resolveremos las dudas que surjan (reuniones con los profesores y reuniones con los instructores). Así mismo, ustedes continuarán adquiriendo experiencia en el uso de tecnologías de educación virtual, lo que les será de provecho para su futuro. Trabajaremos en dos modalidades, una asíncrona en la cual se pondrá a su disposición materiales de estudio de diferentes formatos, y otra síncrona, en la que podrán resolver sus dudas y hacer sus prácticas de laboratorio con apoyo directo de un instructor a cargo. Nuestros encuentros síncronos se realizarán dos veces por semana. Utilizaremos la plataforma Zoom y se les asignará un usuario y contraseña luego de que las autoridades administrativas proporcionen el listado definitivo de estudiantes del curso. Esperamos que su participación sea muy activa a lo largo del semestre en los espacios virtuales que destinemos para ello. Si bien es cierto que no nos citaremos de forma presencial, pero esperamos que cada uno de ustedes ponga de su parte y tenga muchos deseos de aprender. Se evaluará el trabajo individual y las participaciones en las actividades grupales cuando estas se establezcan: habrá evaluaciones en línea durante el semestre, en las que se medirá su progreso acerca de lo aprendido en clase, en sus lecturas y en sus sesiones de Laboratorio. Si tienen alguna pregunta acerca del material del curso pueden escribirnos a nuestros correos electrónicos o a los de los instructores; y si necesitan apoyo técnico acerca de su acceso y uso de las herramientas pueden acudir a los técnicos de Laboratorio al correo **soportefia@uca.edu.sv**.

III. Objetivos de aprendizaje

- Desarrollar la capacidad de pensamiento algorítmico y de implementación de soluciones informáticas a problemas planteados.
-
- Conocer algoritmos y técnicas complejas que le ayudarán a elaborar soluciones eficientes en su trabajo profesional.
-
- Aprender a administrar el recurso memoria RAM para construir programas que requieran de este recurso en tiempo de ejecución.
-
- Desarrollar la capacidad de trabajo colaborativo.

IV. Programación de contenidos y actividades

En esta sección se colocan las temáticas y actividades que estaremos realizando cada semana. Se muestra, en la columna de la derecha, el tiempo estimado que se le debe dedicar a cada actividad.

Semana			Tema	Detalle de actividades a realizar (Tiempo de dedicación)
1	Jueves 12 de agosto	- - Sábado 14 de agosto	Presentación de la materia. Repaso de punteros simples. Punteros dobles y punteros triples.	Inicio de clases jueves 12 de agosto. <ul style="list-style-type: none"> • Sesión síncrona: (60 min). • Revisión de documentación disponible en plataforma (120 min). • Tiempo individual del estudiante (60 min).
2	Lunes 16 de agosto	- - Sábado 21 de agosto	Repaso de estructuras, uniones y clases. Reservar memoria de forma dinámica, comados malloc y new. Liberar memoria, comandos free y delete.	<ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min). • Realización de ejercicios (120 min).
3	Lunes 23 de agosto	- - Sábado 28 de agosto	Implementación de funciones recursivas, conceptos. Tipos de recursión: directa, indirecta, del cola y por posposición.	<ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min). • Realización de ejercicios (120 min).
4	Lunes 30 de agosto	- - Sábado 4 de septiembre	Estructuras de datos, conceptos: Estructuras de datos estáticas vrs. estructuras de datos dinámicas. Manipulación: inserción, búsqueda y eliminación de elementos en estructuras de datos dinámicas.	<ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min). • Realización de ejercicios (120 min).
5	Lunes 06 de septiembre	- - Sábado 11 de septiembre	1a. sub evaluación. Listas, conceptos. Clasificación de las listas en sus tipos fundamentales.	<ul style="list-style-type: none"> • Evaluación (120 min). • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min).
6	Lunes 13 de septiembre	- - Sábado 18 de septiembre	1er. examen parcial. Listas lineales simplemente enlazadas. Manipulación: diferentes formas de inserción de nodos, recorrido y búsqueda.	Asueto miércoles 15 de septiembre. <ul style="list-style-type: none"> • Evaluación (120 min). • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min).

					<ul style="list-style-type: none"> Investigación propia: lecturas, búsquedas (120 min).
7	Lunes 20 de septiembre	- -	Sábado 25 de septiembre	<p>Diferentes formas de eliminación de elementos de una lista simplemente enlazada.</p> <p>Aplicaciones de listas: la pila y la cola.</p> <p>Evaluación de expresiones aritméticas con pilas.</p>	<ul style="list-style-type: none"> Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min). Realización de ejercicios (120 min).
8	Lunes 27 de septiembre	- -	Sábado 2 de octubre	<p>Listas lineales doblemente enlazadas.</p> <p>Manipulación: diferentes formas de inserción de nodos, recorrido y búsqueda.</p>	<ul style="list-style-type: none"> Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min). Realización de ejercicios (120 min).
9	Lunes 04 de octubre	- -	Sábado 9 de octubre	<p>Listas circulares simplemente enlazadas. Manipulación: diferentes formas de inserción de nodos, recorrido y búsqueda.</p> <p>Aplicación: estructura básica de una hoja electrónica y operaciones fundamentales.</p>	<ul style="list-style-type: none"> Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min). Realización de ejercicios (120 min).
10	Lunes 11 de octubre	- -	Sábado 16 de octubre	<p>2a. sub evaluación.</p> <p>Listas circulares doblemente enlazadas. Manipulación: diferentes formas de inserción de nodos, recorrido y búsqueda.</p>	<ul style="list-style-type: none"> Evaluación (120 min). Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min).
11	Lunes 18 de octubre	- -	Sábado 23 de octubre	<p>2o. examen parcial.</p> <p>Estructuras dinámicas de árbol, conceptos.</p> <p>Arboles binarios y árboles binarios de búsqueda: inserción, búsqueda y eliminación de nodos.</p>	<ul style="list-style-type: none"> Evaluación (120 min). Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min).
12	Lunes 25 de octubre	- -	Sábado 30 de octubre	<p>Arboles binarios balanceados, AVL.</p> <p>Operaciones de inserción y eliminación aplicando balanceo.</p>	<ul style="list-style-type: none"> Revisión de documentación disponible en plataforma (120 min). Sesión síncrona: (120 min). Investigación propia: lecturas, búsquedas (120 min). Realización de ejercicios (120 min).

13	Lunes 01 de noviembre	- -	Sábado 6 de noviembre	Arboles multicamino, conceptos. Arboles B, inserción de llaves y aplicación de balanceo.	Asueto martes 2 de noviembre <ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (60 min). • Sesión síncrona: (60 min). • Investigación propia: lecturas, búsquedas (60 min). • Realización de ejercicios (60 min).
14	Lunes 08 de noviembre	- -	Sábado 13 de noviembre	Eliminación de llaves en árboles B y aplicación de balanceo.	<ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min). • Realización de ejercicios (120 min).
15	Lunes 15 de noviembre	- -	Sábado 20 de noviembre	3a. sub evaluación. Arboles B+, inserción de llaves y aplicación de balanceo.	<ul style="list-style-type: none"> • Evaluación (120 min). • Revisión de documentación disponible en plataforma (120 min). • Sesión síncrona: (120 min). • Investigación propia: lecturas, búsquedas (120 min).
16	Lunes 22 de noviembre	- -	Sábado 27 de noviembre	Eliminación de llaves en árboles B+ y aplicación de balanceo.	Ultimo día de clases miércoles 24 de noviembre. <ul style="list-style-type: none"> • Revisión de documentación disponible en plataforma (60 min). • Sesión síncrona: (60 min). • Investigación propia: lecturas, búsquedas (60 min). • Realización de ejercicios (60 min).
17	Lunes 29 de noviembre	- -	Sábado 4 de diciembre	Evaluación final.	<ul style="list-style-type: none"> • Evaluación (120 min).

Resumen de las evaluaciones

Fecha de realización	Evaluación	Porcentaje	Modalidad	Comentarios generales
Martes 7 de septiembre	Primer examen corto	10%	Virtual	
Martes 14 de septiembre	Primer examen parcial	15%	Virtual	
Martes 12 de octubre	Segundo examen corto	10%	Virtual	
Martes 19 de octubre	Segundo Examen Parcial	20%	Virtual	

Viernes 12 de noviembre	Práctica guiada	15%	Virtual	
Martes 16 de noviembre	Tercer examen corto	10%	Virtual	
Lunes 29 de noviembre	Examen Final	20%	Virtual	
		100%		

V. Referencias bibliográficas

Cairó, O., Guardati, S., *Estructuras de datos.*, McGraw Hill, México, 3a. edición.

Joyanes, L., Fernandez, M., Sánchez, L., y Zahonero, I., *Estructuras de datos en C*, McGraw Hill, Espana, 2005.

Deitel y Deitel, C++ cómo programar.

C++ con Clase: <http://c.conclase.net/curso/?cap=000#inicio>

VI. Políticas

Medios de comunicación

El medio de comunicación oficial es el correo electrónico institucional: rcanizales@uca.edu.sv y gcortes@uca.edu.sv. Para situaciones de urgencia pueden marcar al teléfono de la universidad, 2210-6600, en la extensión 1026.

Solicitudes de revisiones

1. Al momento de recibir una evaluación calificada, si el/la estudiante considera que existe un error en la calificación, contará con un plazo de tres días hábiles a partir de la notificación de su calificación para solicitarle al docente, a través de correo institucional, la revisión de la evaluación.
2. El/la estudiante deberá adjuntar a la solicitud las evidencias que respalden la petición de revisión de la evaluación, especificando el apartado o ítemes para los que solicita revisión.
3. A partir de la fecha en que reciba la solicitud, el/la docente contará con un plazo máximo de ocho días hábiles para responder por escrito los argumentos planteados por el/la estudiante y resolver la solicitud.
4. Si el/la estudiante no estuviera conforme con la revisión, contará con tres días hábiles a partir de la notificación de el/la docente para interponer solicitud de revisión en segunda instancia.
5. El/la estudiante enviará a la jefatura del departamento correspondiente un correo institucional solicitando revisión en segunda instancia, adjuntado la resolución de el/la docente y las evidencias que respalden la petición de revisión en segunda instancia, describiendo los apartados o ítems específicos para los que solicita revisión.
6. A partir de la fecha en que se reciba la solicitud, la jefatura del departamento contará con diez días hábiles para resolver.

Solicitud de evaluaciones diferidas.

1. Para solicitar una evaluación diferida el estudiante debe seguir los pasos siguientes:
 - a. Realizar solicitud mediante el sistema de registro de notas (SRN)
 - b. Exponer las razones y adjuntar las pruebas que justifiquen la realización del diferido.
 - c. Cancelar el arancel correspondiente a la solicitud de diferido. El estudiante debe estar solvente con la universidad para solicitar diferido.
 - d. Una vez el diferido sea aprobado, ponerse en contacto con el docente para pactar lugar, fecha y hora de realización de la evaluación diferida.

Anexo – Programa de la asignatura

PROGRAMA DE PROGRAMACIÓN DE ESTRUCTURAS DINÁMICAS

I. GENERALIDADES

Número de orden	: 7
Código:	: 190154
Prerrequisito	: (3) Matemática discreta I (4) Fundamentos de programación
Número de horas por ciclo	: 85
Horas teóricas semanales	: 4
Horas prácticas semanales	: 1
Duración del Ciclo en semanas	: 17
Duración de la hora clase	: 50 min.
Unidades valorativas	: 4
Identificación del Ciclo académico	: II

II. DESCRIPCIÓN

Programación de estructuras dinámicas es una Asignatura teórico-práctica del área de desarrollo de aplicaciones en la que se busca que el alumno asimile las técnicas del desarrollo de programas de computadora, por medio del uso de un lenguaje suficientemente robusto y bajo el paradigma de programación de mayor uso para el desarrollo de aplicaciones en la actualidad. El alumno aprenderá conceptos fundamentales de Programación orientada a objetos y cómo se aplica este paradigma en la creación de programas en el lenguaje C++; tipos de datos estructurados; punteros; aplicación de técnicas recursivas para la solución de problemas en computadora; manejo dinámico de las estructuras de datos fundamentales; y finaliza con la aplicación de comandos y técnicas para el acceso a archivos en disco. Las clases teóricas se complementan con la práctica llevada a cabo semanalmente en los Laboratorios de computadoras en un entorno de Linux y utilizando un compilador de C++. Los alumnos se organizarán en grupos para trabajar en un proyecto final de ciclo en el que elaborarán un programa que resuelva una tarea asignada por su profesor, con lo que se contribuirá al desarrollo de sus habilidades de comunicación y de trabajo en equipo. Estos conocimientos le ayudarán a los estudiantes a aprender a desarrollar programas de computadora robustos, de buen nivel y les desarrollarán la habilidad para aprender otro tipo de lenguajes y herramientas de manera rápida y autodidacta.

III. OBJETIVOS

Objetivo general

Programación de estructuras dinámicas tiene como objetivo principal que el estudiante conozca las estructuras fundamentales de datos y que sepa implementarlas por medio de programas de computadora que resuelvan problemas aplicando técnicas modernas de programación.

IV. CONTENIDO

UNIDAD 1. INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN C++

Objetivo: Conocer los fundamentos del lenguaje de programación C++, su evolución e historia, sintaxis, simbología, operadores y comunicación de datos desde consola.

- 1.1. Historia del lenguaje C++.
- 1.2. Generalidades de los programas escritos en C++.
- 1.3. Operadores.
- 1.4. Entrada y salida estándar.

UNIDAD 2. ESTRUCTURA GENERAL DE PROGRAMAS.

Objetivo: Conocer la organización general que deben tener los programas de computadora que se escriben utilizando el lenguaje C++, división de tareas, comunicación entre procesos, y los sobre la declaración de variables y las instrucciones estructuradas fundamentales.

- 2.1. Modularización.
- 2.2. Paso de parámetros y retorno de resultados.
- 2.3. Declaración de variables y ámbito.
- 2.4. Instrucciones de control de flujo en un programa.

UNIDAD 3. ESCRITURA DE PROGRAMAS BAJO EL PARADIGMA ORIENTADO A OBJETOS.

Objetivo: Adquirir destreza en la elaboración de programas de computadora escritos en C++ utilizando técnicas de Programación orientada a objetos y los recursos propios que este lenguaje posee para el desarrollo de programas utilizando este paradigma de programación.

- 3.1. C++ y la Programación orientada a objetos.
- 3.2. Implementación de clases y objetos.
- 3.3. Métodos.
- 3.4. Atributos de un objeto.

3.5. Encapsulamiento.

3.6. Herencia.

3.7 Manejo de excepciones.

UNIDAD 4. APLICACIONES CON ESTRUCTURAS DINÁMICAS DE DATOS.

Objetivo: aprender el manejo de las estructuras de datos fundamentales que puede implementar para la solución de problemas de programación que manejan conjuntos organizados de datos.

4.1. Listas.

4.2. Árboles.

UNIDAD 5. ACCESO A ARCHIVOS DE DATOS EN DISCO.

Objetivo: comprenda la importancia de mantener almacenes de datos de forma permanente y que aprenda a aplicar técnicas para el acceso externo a disco tanto para la lectura como para la escritura de datos.

5.1. Flujos de salida de un programa.

5.2. Flujos de entrada hacia un programa.

5.3. Acceso secuencial.

5.4. Acceso aleatorio.

V. ESTRATEGIA METODOLÓGICA

El proceso de enseñanza – aprendizaje para este curso, se desarrollará por medio de las siguientes actividades:

Clases teóricas	80%
Trabajo práctico	20%

VI. EVALUACIÓN

La forma de evaluar el aprendizaje en esta asignatura se hará a través de exámenes escritos u orales, trabajos ex aula, exposiciones, laboratorios, u otras actividades consistentes con los objetivos de la asignatura y con el desarrollo del programa en el aula por el docente.

Para lograr esto se sugieren las siguientes evaluaciones y sus porcentajes:

- Exámenes escritos 60%

- Evaluaciones prácticas 20%
- Proyectos 20%

VII. BIBLIOGRAFÍA

1. Deitel, P. J. y Deitel, H. M., *Cómo Programar en C++*, Editorial Pearson Prentice Hall, México, 6ª edición, 2009(3 ejemplares disponibles en Biblioteca).
2. Ceballos Sierra, F. J., *enciclopedia del lenguaje C++*, Editorial Alfaomega Ra-Ma, México, 2ª edición, 2009. (3 ejemplares disponibles en Biblioteca).
3. Cairó, O. y Guardati, S., *Estructuras de datos*, Editorial Mc Graw Hill, México, 3ª edición, 2006. (3 ejemplares disponibles en Biblioteca).
4. Cortés Villeda, Guillermo Ernesto. *Teoría y problemas de programación en C y C++*. 1ª edición. UCA Editores, El Salvador, 2003 (3 ejemplares disponibles en Biblioteca).
5. Dale, Nell. *Programación y resolución de problemas con C++*. 1ª edición. McGraw-Hill Interamericana, México, 2010 (13 ejemplares / Disponible a través de la Biblioteca Digital McGraw-Hill).
6. Joyanes Aguilar, Luis; Zahonero, Ignacio. *Estructura de datos en C++* (Recurso electrónico). McGraw-Hill, España, 2007 (Multiusuario / Disponible a través de la base de datos E-libro).
7. Joyanes Aguilar, Luis. *Programación en C, C++, Java y UML*. 1ª edición. McGraw-Hill, México, 2010 (3 ejemplares disponibles en Biblioteca).
8. Joyanes Aguilar, Luis. *Programación en C++: algoritmos, estructuras de datos y objetos* (Recurso electrónico). 2ª edición. McGraw-Hill, España, 2006 (Multiusuario / Disponible a través de la base de datos E-libro).

SITIOS WEB

<http://c.conclase.net/curso/index.php>