



San Francisco, 22 de diciembre de 2021

VISTO lo dispuesto por la Ordenanza 1383/12, y

CONSIDERANDO:

Que por medio de esta normativa y mediante el dictado de asignaturas electivas es posible incorporar perfiles propios de la región a efectos de adaptar los diseños curriculares a las necesidades de la misma.

Que en tal sentido y en cumplimiento de las reglamentaciones vigentes, y a propuesta de los Departamentos respectivos los Consejos Directivos de las Facultades Regionales definirán cuáles serán las materias electivas, área del conocimiento, objetivos generales y específicos que justifiquen la inclusión, carga horaria, sus contenidos analíticos, bibliografía, modalidad de dictado, propuesta pedagógica, y sus correspondientes correlatividades debidamente justificadas.

Que el Consejo Departamental de Ing. en Sistemas de Información elevó al Consejo Directivo de esta Facultad Regional San Francisco la propuesta de implementación de materias electivas.

Que la Comisión de Enseñanza evaluó la propuesta emitiendo despacho favorable.

Que el dictado de la medida se efectúa en uso de las atribuciones otorgadas por el Estatuto Universitario.

Por ello,

**EL CONSEJO DIRECTIVO DE LA FACULTAD REGIONAL SAN FRANCISCO
DE LA UNIVERSIDAD TECNOLÓGICA NACIONAL
RESUELVE**

ARTÍCULO 1°.- Aprobar la continuidad del dictado de la asignatura Construcción de Software (carga horaria anual 4 hs.) como materia electiva, parte curricular de la Carrera Ingeniería en Sistemas de Información del área Programación a dictarse en el tercer nivel, con modalidad cuatrimestral (segundo cuatrimestre) y una carga horaria de 8 horas semanales.

ARTÍCULO 2°.- Aprobar en Anexo I, Objetivo General y objetivos específicos que justifican la inclusión de dicha materia, las correlatividades debidamente justificadas, el programa analítico, la bibliografía y la propuesta pedagógica.

ARTÍCULO 3°.- Regístrese. Comuníquese. Elévese al Rectorado a sus efectos y archívese.

RESOLUCIÓN CD N°: 535/2021



Ing. JUAN CARLOS GALLONI
Secretaría Académica

Firma Digital

Aprobación del Documento por Juan Carlos Calloni
UNIVERSIDAD TECNOLÓGICA NACIONAL FR SAN FRANCISCO



Ing. Alberto R. TOLOZA
Decano

Firma Digital

Aprobación del Documento por Alberto Toloza
UNIVERSIDAD TECNOLÓGICA NACIONAL - FR SAN FRANCISCO



Construcción de Software

Área: Programación

Carga horaria: 8 horas semanales.

Modalidad de cursado: Cuatrimestral - 2do cuatrimestre.

1. Objetivos generales y específicos que justifican la inclusión de la Materia

Objetivo General:

Introducir en el análisis y la evaluación de buenas costumbres de escritura de código de software, brindando a los estudiantes las mejores técnicas y métodos.

Objetivos específicos:

- I. Escribir código con buenas costumbres de programación.
- II. Saber distinguir un código escrito siguiendo criterios de calidad.
- III. Adquirir y evaluar conceptos tales como la programación defensiva y ocultamiento de información.
- IV. Asumir la importancia de programar pensando en la calidad del código fuente.
- V. Comprender la necesidad de ser disciplinado en la programación de software y en el uso de sus herramientas.

2. Correlatividades debidamente justificadas

Para Cursar

Regularizadas

- a. **Gestión de Datos:** Esta materia es necesario tenerla cursada y regular ya que el estudiante necesita tener hábitos en la descripción de algoritmos para el tratamiento de la información almacenada, conocer la utilización de tipos de accesos diferentes, expresiones e instrucciones, conocer el diseño de algoritmos, poniendo en práctica las distintas operaciones para consultar, actualizar y mantener la información guardada en los archivos.
- b. **Análisis de Sistemas:** Esta materia es necesario tenerla cursada y regular ya que el estudiante necesita saber modelar las características intrínsecas de los sistemas de información, conocer y aplicar las metodologías, modelos, técnicas y lenguajes de la etapa de análisis, seleccionar adecuadamente los modelos que mejor se adapten para dar soluciones a los problemas de información.

Aprobadas

- a. **Algoritmos y estructuras de datos:** Esta materia es necesario tenerla aprobada ya que el estudiante necesita tener afianzados los contenidos para poder resolver situaciones problemáticas bajo el paradigma imperativo (representación gráfica,



Ing. Gabriel Cerutti

Dir. Dpto.



lenguaje nemotécnico), organizar los programas: paradigmas, lenguajes, desarrollos y estilos, tener los conocimientos para almacenar datos en las distintas estructuras de datos, como una manera conceptual de organizar los datos (pilas, colas, tablas, listas, arboles, grafos).

Para Rendir

Regularizadas

- a. **Diseño de Sistemas:** Esta materia es necesario tenerla cursada para rendir la cátedra en cuestión ya que el estudiante requiere disponer de las herramientas necesarias para el diseño de un sistema de información y su exitosa construcción, conocer las metodologías, modelos, técnicas y lenguajes del proceso de diseño, así como también el diseñar y construir productos de software asociado a los sistemas de información aplicando herramientas de soporte de diseño.

Aprobadas

- a. **Gestión de Datos:** Esta materia es necesario tenerla aprobada para rendir la cátedra en cuestión, ya que el estudiante necesita tener afianzados los contenidos para conocer la utilización de tipos de accesos diferentes, expresiones e instrucciones, conocer el diseño de algoritmos, poniendo en práctica las distintas operaciones para consultar, actualizar y mantener la información guardada en los archivos. Asimismo es importante que conozca sobre el diseño de bases de datos y forma normales.
- b. **Análisis de Sistemas:** Esta materia es necesario tenerla aprobada para rendir, ya que el estudiante necesita tener afianzados los contenidos referidos a características intrínsecas de los sistemas de información, conocer y aplicar las metodologías, modelos, técnicas y lenguajes de la etapa de análisis, seleccionar adecuadamente los modelos que mejor se adapten para dar soluciones a los problemas de información

3. PROGRAMA ANALÍTICO

EJE TEMÁTICO Nº 1: Prerrequisitos de la construcción

Introducción a los prerrequisitos de la construcción de software. Necesidad de los distintos prerrequisitos. Identificación de los elementos fundamentales en cada prerrequisito.

EJE TEMÁTICO Nº 2: Diseño

Pasos en la construcción de una rutina, herramientas a utilizar. Principios a aplicar en la codificación. Características de rutinas de alta calidad.

EJE TEMÁTICO Nº 3: Datos

Tipos de datos. Principios a utilizar en la creación de variables. Convenciones para nombres. Creación de tipos de datos. Tipos e inicialización de variables. Nombres de variables y convenciones para la definición de su nombre.

EJE TEMÁTICO Nº 4: Control

Estructuras de control. Organización del código. Estructuras de control inusuales. Expresiones booleanas. Buenas formas de utilización en las estructuras de control. Anidamientos profundos. Reducción de complejidad en las expresiones booleanas.



EJE TEMÁTICO Nº 5: Consideraciones constantes

Estilo de programación como documentación del código. Técnicas para comentarios. Tamaño de código y dificultades en su construcción. Tipos y estilos de comentarios. Efectos del tamaño en el desarrollo y en los errores.

EJE TEMÁTICO Nº 6: Mejorar la calidad

Características de la calidad del software. Técnicas para mejorar la calidad del software. Revisiones e inspecciones.

4. BIBLIOGRAFÍA

OBLIGATORIA:

- FOWLER, Martin.
Refactoring: Improving the Design of Existing Code
Pearson Education. 2019.
- McCONNELL, Steve.
Code Complete 2. A PRACTICAL HANDBOOK OF SOFTWARE CONSTRUCTION.
Microsoft press. 2004.
- GOMEZ BLANES, Rafael.
El libro negro del programador. Cómo conseguir una carrera de éxito desarrollando software y cómo evitar los errores habituales
Edit. Titivillus. 2004.
- READ, Robert.
How to Be a Programmer: A Comprehensive Summary
CreateSpace Independent Publishing Platform. 2009.

5. PROPUESTA PEDAGÓGICA

El nuevo material de aprendizaje debe relacionarse significativamente, para integrarse en su estructura cognoscitiva previa, modificándola y produciendo un conocimiento duradero y sólido. Si se producen aprendizajes verdaderamente significativos, se consigue uno de los objetivos principales de la educación: asegurar la funcionalidad de lo aprendido. Es necesario abordar los aspectos teórico-prácticos como forma de generación de conocimiento, considerando dicha práctica como praxis y no como aplicación.

La metodología de trabajo comprende:

Clases en aula y en el laboratorio.



Las clases teóricas estarán a cargo del profesor. En las mismas se impartirán los conocimientos claves del contenido de materia, presentando y evaluando buenas costumbres de programación destinadas a lograr código de buena calidad. Se ejemplificarán la los conceptos en diferentes lenguajes de programación.

La teoría será reafirmada mediante la realización de cuestionarios que los alumnos deberán responder en grupo en clase

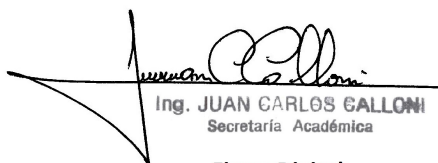
Práctica y laboratorio

Estarán a cargo del profesor y los auxiliares que la catedra pueda disponer. En las mismas se impartirán guías de ejercicios. Se codificarán rutinas y programas en el lenguaje de programación que el alumno considere apropiado (teniendo en cuenta la disponibilidad de lenguajes en el laboratorio). Los trabajos prácticos se realizarán siguiendo el desarrollo de aplicaciones simples en las cuales el estudiante deberá ir aplicando los conocimientos adquiridos en la teoría y práctica.

Recursos metodológicos:

Para el desarrollo de las distintas actividades de la asignatura se utilizan los siguientes materiales didácticos:

- Pizarra y fibra
- Pantalla y Cañón de proyección
- Acceso a Internet
- Campus Virtual
- Laboratorio de computación



Ing. JUAN CARLOS CALLONI
Secretaría Académica

Firma Digital

Aprobación del Documento por Juan Carlos Calloni
UNIVERSIDAD TECNOLÓGICA NACIONAL FR SAN FRANCISCO



Ing. Alberto R. TOLOZA
Decano

Firma Digital

Aprobación del Documento por Alberto Toloza
UNIVERSIDAD TECNOLÓGICA NACIONAL - FR SAN FRANCISCO