



Facultad de Ciencias
de la **Administración**

TECNICATURA
UNIVERSITARIA EN
**DESARROLLO
WEB**



PROGRAMACIÓN I

Unidad I – Introducción a la programación de computadoras

Ciclo de vida de Desarrollo de Software

2022 - Primer Cuatrimestre

Tecnicatura Universitaria en Desarrollo Web

Facultad de Ciencias de la Administración

Universidad Nacional de Entre Ríos

Unidad I – Introducción a la programación de computadoras

- **Objetivos**

- Comprender los conceptos de algoritmo, programa y lenguaje de programación.
- Conocer e identificar cada una de las etapas del ciclo de vida de desarrollo de software.

- **Temas a desarrollar:**

- **Algoritmos y programas**

- Programa. Concepto. Algoritmos. Concepto, características.
- Computadoras. Características y Componentes.

- **Lenguajes de programación**

- Definición y Niveles.

- **Paradigmas de programación**

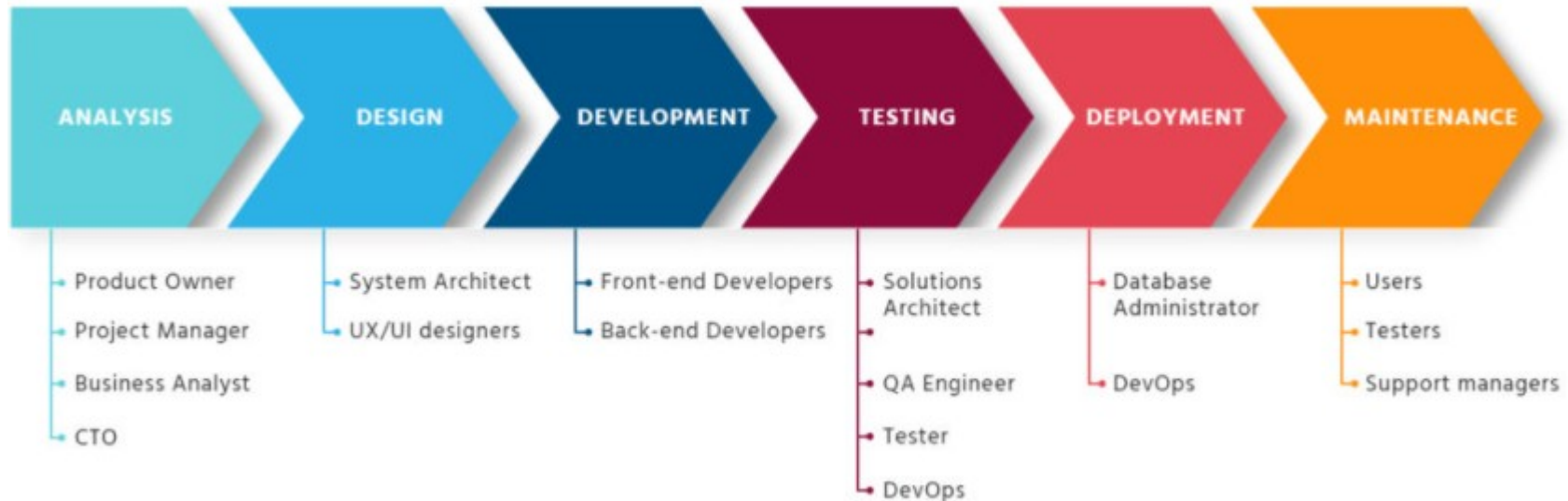
- Definición y Clasificaciones.

- **Ciclo de vida de desarrollo de software**

- Etapas.

Ciclo de vida de desarrollo de software

- El **proceso de desarrollo** o **ciclo de vida del software (SDLC)** es una secuencia estructurada y bien definida de **etapas** que deben seguirse para desarrollar un producto software. Es una de las ramas de estudio de la **Ingeniería de Software**.
- Existen distintos modelos que describen el proceso de desarrollo, los cuales plantean distintas **etapas** y formas de proceder para que un equipo de programadores lleve a cabo el desarrollo de un programa.



Modelo en cascada

- El **modelo de desarrollo en cascada** es el más clásico, el primero en aparecer y a partir del cual surgen las técnicas más modernas. En este modelo se plantea seguir una serie de etapas de manera secuencial, las cuales se detallan a continuación:
- **Análisis de requisitos**
 - Se analiza la problemática a resolver y las necesidades de los futuros usuarios del programa, convirtiéndolos en requisitos concretos que debe cumplir el programa.

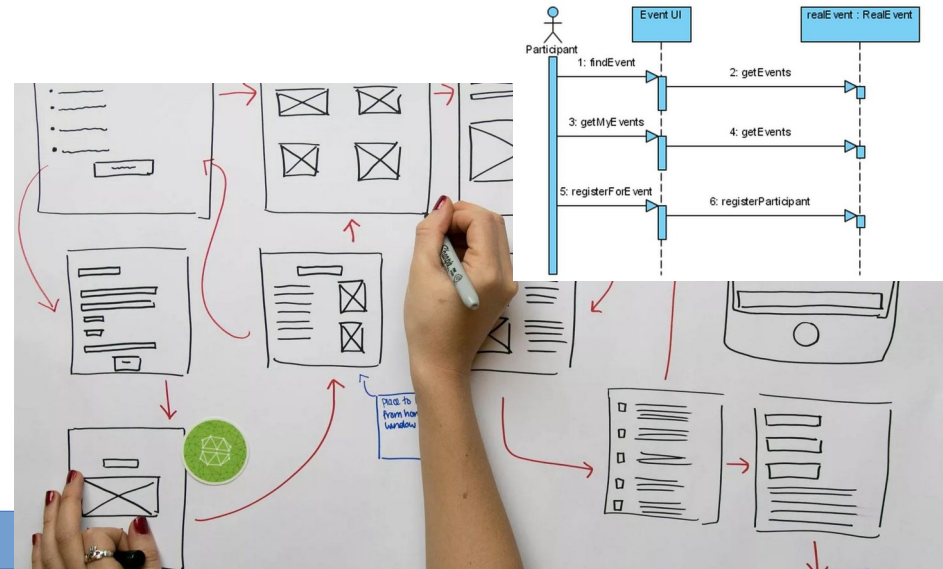
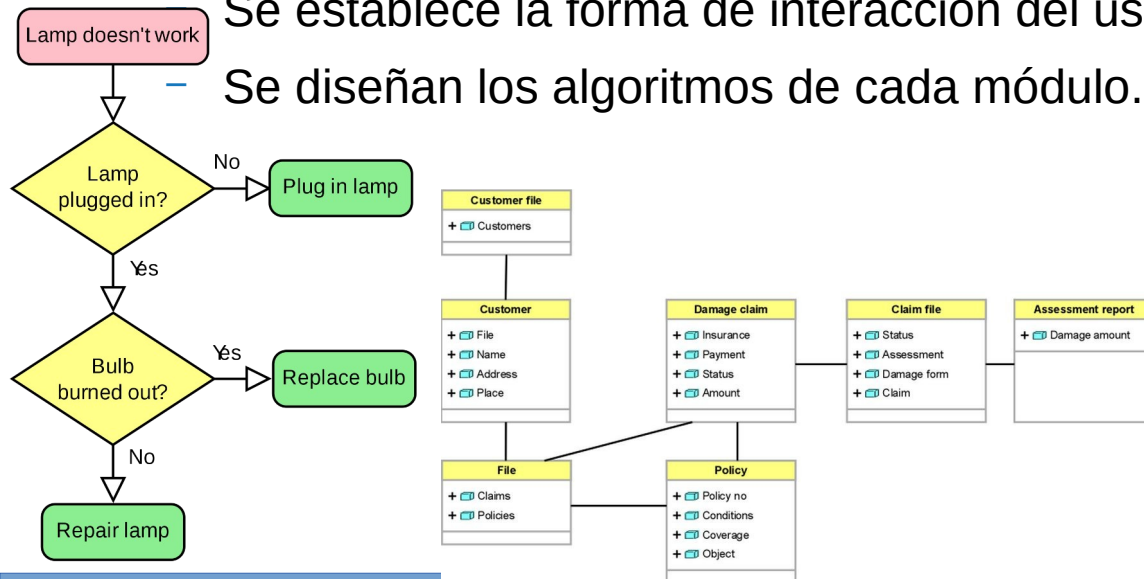


Modelo en cascada (2) – Diseño de la solución

- En esta etapa se estudian posibles **opciones** de **implementación** para el software que hay que construir, así como decidir la estructura general del mismo.
- Se trata de una etapa compleja y su proceso debe realizarse de manera iterativa.
- Se realizan **diagramas** de las distintas partes del software (módulos, funciones).
- Se determina la manera en que se representará y manejará la información de interés (variables, constantes, estructuras de datos).

Se establece la forma de interacción del usuario (interfaz, entradas y salidas).

- Se diseñan los algoritmos de cada módulo.



Modelo en cascada (3) – Codificación y pruebas unitarias

- En esta etapa se debe elegir las **herramientas adecuadas**, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir.
 - Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.
- Se escribe el **código fuente** de los algoritmos de los distintos módulos en un lenguaje de programación y se **documentan** los componentes del software.
- En la medida que se van construyendo los componentes se prueban de manera aislada. Esto puede ser programando **pruebas automatizadas** o haciéndolas manualmente.
 - Se controla que las salidas sean las esperadas para cada una de las diferentes entradas.

Modelo en cascada (4)

- **Integración, validación y verificación del programa**

- Se **integran** los distintos módulos ya depurados, conformando el programa ejecutable.
- Se ejecuta el programa y se **verifica** el cumplimiento de los distintos requisitos generados en la fase de análisis.
- Se **corrigen** los **errores** encontrados.

- **Implementación o despliegue**

- Implica todas las actividades que hacen que un software esté disponible para su uso.
- Se instala el software en los equipos que corresponda (servidores, dispositivos y/o PCs usuario).
- Puede requerir la instalación y/o configuración de software adicional.
- La capacitación y soporte de usuarios es de suma importancia en esta etapa.

- **Mantenimiento**

- Corrección de fallas detectadas durante el uso y de necesidades cubiertas parcialmente.
- Se implementan mejoras de eficiencia.

- **Sobre la aplicación del modelo en cascada**

- Seguir a rajatabla el modelo en cascada **no es eficiente**, especialmente cuando esperamos concluir toda la fase de codificación para comenzar las pruebas. Lo cual retrasa la detección de errores de funcionamiento y dificulta su identificación en el código.
- Suele ser conveniente entremezclar estas dos fases, de manera tal de ir codificando, ejecutando y probando de a partes pequeñas de código. Así es posible identificar con más facilidad las secciones de código con errores y solucionarlos a la brevedad.

- **Modelo repetitivo**

- Este modelo guía el proceso de desarrollo de software en repeticiones. Así, proyecta el proceso de desarrollo de modo cíclico repitiendo cada paso después de cada ciclo en el proceso de ciclo de vida del software.

- **Modelo en espiral**

- El modelo en espiral es una combinación de los modelos anteriores donde se tiene en cuenta el riesgo. De esta forma, se comienza fijando los objetivos y las limitaciones al empezar cada repetición. En la etapa siguiente se crean los modelos de prototipo del software, que incluye el análisis de riesgo. Posteriormente se usa un modelo estándar para construir el software y finalmente se prepara el plan de la próxima repetición.

- **Modelo en V**

- Uno de los grandes problemas del modelo en cascada es que solo se pasa a la siguiente fase si se completa la anterior y no se puede volver atrás si hay errores en etapas posteriores. Así, el modelo en V da más opciones de evaluación del software en cada etapa.
- En cada fase se crea la planificación de las pruebas y los casos de pruebas para verificar y validar el producto en función de los requisitos de la misma. De esta manera, verificación y validación van en paralelo.

Bibliografía

- Pablo A. García, Marcelo A. Haberman, Federico N. Guerrero: ***“Programación E1201: curso de grado”***. 1Era Edición. Ed. Editorial de la UNLP. 2021.
- Óscar Ramírez Jiménez: ***“Python a fondo”*** 1era Edición. Ed. Marcombo S.L.. 2021.
- Allen Downey. ***“Think Python”***. 2Da Edición. Green Tea Press. 2015.
- Eirc Matthes: ***“Python Crash Course”***. 1era Edición. Ed. No Starch Press. 2016.
- Zed A. Shaw: ***“Learn Python 3 the Hard Way”***. 1era Edición. Ed. Addison-Wesley. 2017.
- Armando E. De Giusti.: ***“Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci”***. Ed. Buenos Aires Prentice Hall; Pearson Educación. 2001.