

# TRABAJO FINAL INTEGRADOR

PROGRAMACIÓN 1 - 2022 - 1er cuatrimestre  
TECNICATURA UNIVERSITARIA EN DESARROLLO WEB

## OBJETIVOS

Son objetivos de este Trabajo Final Integrador que el estudiante:

- Ponga en práctica todos los conocimientos adquiridos durante el cursado de la asignatura:
- Tenga una experiencia cercana a desarrollar un programa real con requerimientos establecidos por el/los usuario/s o encargados del Análisis de Requerimientos.
- Tome decisiones de diseño de un programa.

## CONDICIONES DE ENTREGA

- El Trabajo Final Integrador deberá ser:
  - Realizado en forma individual o en **grupos de NO más de 4 (cuatro) alumnos**.
  - Cargado en la sección del Campus Virtual correspondiente, en un archivo comprimido en formato ZIP, RAR, TAR.GZ u otro formato similar.
  - En caso de realizar el Trabajo Final Integrador en grupo, deberá indicarse el apellido y nombre de los integrantes del mismo.
  - Entregado antes de la fecha límite informada en el campus.
- El/los integrantes del equipo deberán filmar un **video** donde realicen una exposición del proyecto. En el mismo se deberá detallar el funcionamiento del programa y las decisiones de diseño tomadas para su construcción. El **video** debe:
  - Tener audio y en el mismo debe mostrarse la pantalla del dispositivo donde se visualice el código y el programa en ejecución y al menos la voz en off de alguno de los integrantes con las explicaciones. Idealmente hacer captura en video del/los integrantes haciendo la explicación.
  - Ser cargado a YouTube, Vimeo u otra plataforma similar y el enlace para su visualización entregado junto con los archivos que componen el proyecto.
- El Trabajo Final Integrador será calificado con nota numérica y determinará si el/los alumno/s promociona/n o no la asignatura.
- Las soluciones del alumno/grupo deben ser de autoría propia. Aquellas que se detecten como idénticas entre diferentes grupos serán clasificadas como **MAL** para todos los involucrados en esta situación que será comunicada en la devolución.
- Se valorarán la exactitud, eficiencia y prolijidad (identación y otras buenas prácticas) de las soluciones planteadas, así como también la calidad de la exposición realizada.

## ENUNCIADOS

A continuación se detallan las opciones de implementación el alumno/grupo deberá elegir una de ellas.

## OPCIÓN N.º 1:

La empresa de desarrollo de software en la que Ud. se desempeña ha decidido programar una solución de software del tipo historia clínica para el “*Instituto Médico Las Luciérnagas*”. Realizado el análisis de requerimientos por sus compañeros de equipo, se ha determinado necesario registrar los siguientes datos:

De los **Pacientes**:

- Documento de Identidad,
- Apellido,
- Nombre,
- Fecha de Nacimiento,
- Nacionalidad.

Cada paciente tendrá asociada una **Historia Clínica**, la que contará con los siguientes datos:

- Fecha,
- Enfermedad/afección que padece,
- Médico del instituto que lo trató,
- Observaciones.

Los **Profesionales** que atienden en la clínica deberán registrarse en un archivo diferente con los siguientes datos:

- Apellido,
- Nombre,
- Especialidad.

## REQUERIMIENTOS

- La información de **Pacientes** e **Historias Clínicas** debe ser registrada en archivos en formato JSON.
- Se debe contar con interfaces de usuario interactivas que permitan acceder a las siguientes opciones:
  - En referencia a los **Pacientes**:
    - Registrar uno nuevo.
    - Editar los datos de uno existente.
    - Eliminar un paciente.
  - En relación a las **Historias Clínicas**:
    - Agregar a una historia clínica a un paciente.
  - Funciones de búsqueda de Pacientes a partir de los siguientes criterios:
    - Apellido y/o Nombre,
    - Rango de fechas en la que fueron atendidos (Fecha de Historia clínica),
    - Enfermedad/afección,
    - Por Médico que lo/la trató,

- Nacionalidad
- En cada oportunidad que se muestren los datos de un paciente se deberán calcular su edad a partir de su fecha de nacimiento y fecha actual del sistema.
- Cada nuevo paciente tendrá asignado un número de Historia Clínica. Se trata de un número incremental que inicia en 1.
- De los **médicos**:
  - Permitir listar los existentes y registrar nuevos.

## OPCIÓN N.º 2:

La empresa de desarrollo de software en la que Ud. se desempeña ha decidido programar una solución de software Emisión y Consulta de Comprobantes para el comercio “La Nave Nodriz”, que se dedica a la venta de artículos e insumos informáticos. Realizado el análisis de requerimientos por sus compañeros de equipo, se ha determinado necesario registrar los siguientes datos:

- **Comprobantes:**
  - Fecha,
  - Tipo de Comprobante,
  - Punto de Venta,
  - Número de Comprobante,
  - Importe Total,
  - **Cliente:**
    - Apellido,
    - Nombre,
    - CUIT/CUIL,
    - Domicilio,
  - **Detalle (al menos uno por comprobante):**
    - Número de Ítem,
    - Producto,
    - Cantidad,
    - Precio Unitario,
    - Total (Cantidad \* Precio Unitario)

## REQUERIMIENTOS

- La información debe ser registrada en un único archivo en formato JSON.
- Se debe contar con interfaces de usuario interactivas que permitan acceder a las siguientes opciones:
  - En referencia a los **Clientes**:

- Registrar uno nuevo.
- Editar datos de uno existente.
- Eliminar un cliente.
- Deberán mantenerse en un archivo diferente al de los Comprobantes.
- En referencia a los **Comprobantes**:
  - Permitir registrar nuevos Comprobantes teniendo en cuenta que:
    - Su tipo puede ser Factura A, Factura B o Factura C,
    - El punto de venta es un número de 4 cifras que se completa con ceros a la derecha.
  - El número de comprobante es un número de 8 cifras que se completa con ceros a la derecha.
  - Los datos tipo, punto de venta y número de comprobante no pueden repetirse.
  - El **Importe Total** deberá calcularse a partir de la suma del Total de cada uno de los detalles.
- En referencia al **Detalle** del comprobante:
  - El **número de ítem** es un número incremental para cada comprobante,
  - El **Producto** es solamente una descripción en formato texto.
  - El **Total** no será consignado por el usuario sino que se calculará de multiplicar la cantidad por el precio unitario.
  - No permitir importes en 0 o en negativo para el **precio unitario**.
- Listados. Se deberán poder aplicar los siguientes filtros de información:
  - Listado de Comprobantes por Cliente.
  - Listado de Comprobantes por rango de fechas.
  - Listado de Comprobantes por producto.

## OPCIÓN N.º 3:

La empresa de desarrollo de software en la que Ud. se desempeña ha decidido programar un juego de preguntas y respuestas llamado *“Los 8 escalones de la Programación”*. Realizado el diseño del mismo se determina necesario registrar los siguientes datos:

- **Preguntas:**
  - Enunciado,
  - Escalón (Número del 1 al 8),
  - Ponderación,
  - Respuesta opción N.º 1,
  - Respuesta opción N.º 2,
  - Respuesta opción N.º 3,
  - Respuesta opción N.º 4,
  - Respuesta Correcta.

- **Partidas:**
  - Nombre de jugador,
  - Fecha y Hora en la que tuvo lugar,
  - Preguntas que le tocaron:
    - Opción seleccionada,
    - Es respuesta correcta (bool)
  - Puntaje obtenido.

## REQUERIMIENTOS

- Las preguntas deben ser almacenadas en un archivo JSON y las partidas en otro.
- Se debe contar con interfaces de usuario interactivas que permitan acceder a las siguientes opciones:
  - Modo **“Carga de datos”**:
    - En este modo el usuario podrá registrar nuevas preguntas.
    - Todas las preguntas tienen cuatro opciones, una de las cuales es correcta.
    - Cada pregunta tiene asociada un escalón (número entero de 1 a 8).
  - Modo **“Jugar”**:
    - En este modo el usuario inicia una nueva partida.
    - Comienza en el escalón N.º 1 y con cada respuesta correcta avanza un escalón hasta llegar al escalón N.º 8.
    - En el escalón N.º 8 deberá contestar correctamente 4 preguntas de manera consecutiva para ganar, caso contrario perderá.
    - Al finalizar el juego, se mostrará el puntaje final obtenido que surge de sumar la ponderación de cada pregunta.
  - Modo **“Consultas”**:
    - En este modo se podrán aplicar filtros de partidas por los siguientes criterios:
      - Partidas por jugador.
      - Partidas por rango de fechas.
      - Siempre las partidas deberán mostrarse en orden descendente de puntaje.