

Nombre: _____ Carné _____

Indicaciones Generales

- Fecha de entrega: **martes 11 de julio 10:00 pm.**
- Lea con atención y asegúrese de entender lo que se le solicita antes de responder.
- El examen es individual. No se permite trabajar con otros(as) estudiantes durante el examen. Si se detecta que el(a) estudiante violenta esta indicación, el examen le será anulado y se aplicará la normativa del Reglamento de Orden y Disciplina de los Estudiantes de la UCR.

Parte 0 – Tema a abordar en el examen

El primer paso es elegir el tema que va a utilizar para el examen. Diríjase a la [hoja compartida de temas](#). Escoja uno de los 30 temas colocando su nombre en la columna H. Solo debe escoger un tema. Si la celda ya tiene un nombre o algún (a) compañero(a) está escribiendo en la celda, favor escoger otro tema ya que todos son equivalentes. Evite sobre escribir el tema de otra persona ya que usaremos el historial de cambios en caso de que se tenga que resolver un conflicto. El tema no es determinante para el examen.

En el examen usted va a diseñar una aplicación web pequeña que permita realizar las operaciones conocidas como CRUD (*create, read, update, delete*) sobre una base de datos que contenga registros sobre el tema que eligió anteriormente.

En el tema que eligió se incluyen algunos atributos generales que puede utilizar, así como tipos de datos. No es estrictamente obligatorio apegarse a estos atributos, puede variar algunos si así lo desea. La única restricción es que utilice al menos cinco (5) atributos y al menos los tipos de datos de texto, numérico y booleano.

Parte 1 – Diseño y Desarrollo (50%)

Cree una aplicación web que permita realizar las operaciones conocidas como CRUD (*create, read, update, delete*) para una base de datos (BD) que contenga registros sobre el tema que eligió anteriormente.

La aplicación web debe utilizar el patrón Modelo-Vista-Controlador como lo ha hecho en los laboratorios del curso. Se espera que la BD tenga una o dos tablas como máximo y puede crearlas en el servidor que le fue asignado en el curso de BD. La aplicación web puede ser de una sola página y debe mostrar un diseño agradable para el usuario mediante la aplicación de hojas de estilo. No es necesario que la aplicación incluya registro de usuarios ni sesiones.

10% Correcto uso del patrón de arquitectura MVC

5% Correcto uso de la base de datos

5% Calidad de la interfaz de usuario de la aplicación web (identificación y uso de los servicios sin necesidad de tener conocimiento previo de la aplicación, separación de lenguaje de marcado HTML y estilo CSS)

5% Calidad de la interacción con el usuario (que permita la utilización de los servicios con el menor esfuerzo y sin inducir a errores)

10% Completitud de las cuatro operaciones CRUD (correcto funcionamiento)

3% Control de errores y excepciones (a lo interno del código fuente incluyendo mensajes claros al usuario)

3% Buen uso de *commits* en el repositorio (frecuentes, con sentido y con comentarios claros)

6% Nivel de cumplimiento del código fuente de todos los principios de código limpio (*Clean Code*)

3% Entregar el script de la creación de la base de datos utilizada

Parte 2 – Pruebas Unitarias (20%)

Cree un proyecto de pruebas unitarias en su *framework* de preferencia. Elija cuatro funciones/métodos de su código y cree al menos dos pruebas unitarias para cada una de las funciones utilizando la técnica de particiones en clases de equivalencia. Para cada uno de sus casos de prueba, agregue un pequeño comentario en el código con los siguientes dos elementos:

- 1) justificación breve de la selección de los valores de entrada para esa prueba
- 2) justificación del resultado esperado en sus propias palabras

2% Validez del proyecto de pruebas unitarias (como proyecto de pruebas en el *framework* elegido)

4% Cumplimiento de la cantidad de pruebas unitarias (4 funciones/métodos, 2 pruebas para c/u)

4% Validez de las pruebas unitarias (que sean pertinentes y creadas con fundamento teórico)

2% Resultado de las pruebas unitarias (todas deben pasar al ser ejecutadas)

4% Validez de la justificación de los valores de entrada para cada prueba (con fundamento teórico)

4% Validez de la justificación del resultado esperado para cada prueba (con fundamento teórico)

Parte 3 – Pruebas funcionales automatizadas (25%)

Cree un proyecto de pruebas funcionales automatizadas utilizando la herramienta Selenium WebDriver o cualquier otra de su preferencia que sea equivalente. Para cada una de las operaciones CRUD, cree una prueba automatizada que demuestre el correcto funcionamiento de esa operación. El proyecto debe permitir un registro de los resultados de sus pruebas funcionales. Para cada caso de prueba, agregue un pequeño comentario en el código con los siguientes dos elementos:

- 1) justificación breve del objetivo de la prueba
- 2) justificación breve de cómo el resultado esperado cumple el objetivo de la prueba

1% Validez del proyecto de pruebas funcionales (como proyecto de pruebas en el *framework* elegido)

2% Uso correcto de la herramienta de automatización de pruebas

4% Cumplimiento de la cantidad de pruebas funcionales (4 operaciones, 1 prueba para c/u)

4% Validez de las pruebas funcionales (que sean pertinentes y creadas con fundamento teórico)

4% Resultado de las pruebas funcionales (todas deben pasar al ser ejecutadas)

4% Validez de la justificación del objetivo para cada prueba (con fundamento teórico)

2% Validez de la justificación del resultado esperado para cada prueba (con fundamento teórico)

4% Uso correcto del patrón *Page-Object-Model* en las pruebas funcionales

Parte 4 - Entregables (5%)

Primera parte: Entregar el código de la aplicación, y los dos proyectos de pruebas en su repositorio **PRIVADO** de control de versiones que ha utilizado durante el curso para los laboratorios y al cual el docente tiene acceso.

Segunda parte: un archivo con el nombre **SU#CARNE.PDF** que tenga adentro su nombre y número de carné universitario, así como la ruta del repositorio privado donde se encuentran todos los artefactos generados en el examen. Este documento lo debe subir a Mediación Virtual antes de la hora de cierre del examen.

2% Ubicación clara de todos los artefactos del examen en directorios identificables, por ejemplo:

- a) El código de la aplicación web - **app_source/**
- b) El código del proyecto de pruebas unitarias - **unit_tests/**
- c) El código del proyecto de pruebas funcionales automatizadas - **functional_tests/**

0.5% En un directorio llamado **unit_test_results/** entregue uno o más *screenshots* que muestren el resultado positivo (**pass**) de las pruebas unitarias.

0.5% En un directorio llamado **functional_test_results/** entregue uno o más *screenshots* que muestren el resultado positivo (**pass**) de las pruebas funcionales automatizadas.

2% Instrucciones claras para correr los tres proyectos (aplicación, pruebas unitarias, pruebas funcionales). Puede agregarlas al README.md del repositorio o al archivo PDF que entrega.