

Proyecto Final

SIMULADOR VIRTUAL

Generalidades

Fecha de Entrega: Viernes 25-Febrero-2022

Modalidad: Individual

Objetivo

Aplicar los conocimientos relacionados a la orientación a objetos, entrada y salida, operaciones aritméticas, paso de parámetros y estructuras de control del flujo de programas en un problema de simulación de baja complejidad.

Situación

En el área de investigación de operaciones típicamente se utilizan modelos matemáticos para analizar problemas relacionados a la toma de decisiones bajo restricciones impuestas por variables como tiempo y recursos. Desde el punto de vista de la computación es posible construir programas, que, mediante la solicitud de valores asociados a ciertos parámetros o variables, puedan llevar a cabo simulaciones tendientes a sugerir posibles alternativas para la toma de decisiones.

Problema

Se debe crear un programa web de .Net Core que, mediante la programación orientada a objetos, simule la existencia de dos máquinas y un simulador que permita elegir la máquina que ofrezca más ganancias en un sistema de manufactura con un producto cualquiera. Suponga que todas las máquinas producen el mismo tipo de "Producto", cada producto tiene un nombre y un precio, que debe ser asignado por el usuario. Cada máquina tiene una cantidad de producto que puede fabricar por hora. Además, cada máquina tiene un costo en colones asociado al costo de operación por hora. A cada una se le asocia una probabilidad (valor entre 0.0 y 1.0) de fallar. Todos los datos relativos, deben ser almacenados en la base de datos, conectada al proyecto. Debido a lo anterior, el proveedor de cada máquina le asigna una garantía de reparación de fallas que establece el tiempo máximo en horas que durará la máquina para ser reparada o remplazada sin incurrir en multas (de esta forma solo los días de falla dentro de este período afectan el costo, ya que la multa por reparación tardía corresponderá a la pérdida diaria asociada a los productos que debieron producirse en el tiempo, razón por lo que las multas no se tomarán en cuenta).

Note que cualquiera de las máquinas en un momento dado puede estar funcionando o puede estar descompuesta. La simulación que se quiere llevar a cabo necesita obtener del usuario varios parámetros de configuración como lo son: la cantidad de horas por día que funcionarán las máquinas, cantidad de días por semana que trabajarán las máquinas, cantidad de (meses, días y horas) de producción continua que serán simulados para tomar la decisión, precio por producto y las características de cada máquina incluyendo su costo por hora, la probabilidad de fallar, la cantidad de horas que duran en ser reparada y el estado funcional de la misma. Todos los registros, deben de guardarse en la base de datos, y poseer una vista para ver los resultados de la simulación.

Guía de Implementación

Se asumen varias cosas:

- La simulación llevará a cabo una iteración por unidad de tiempo.
- La unidad de tiempo será de 1 hora, oh sea un valor entero de 60.
- Un mes corresponde a 4 semanas.
- Si una máquina falla, será reparada automáticamente cuando se llegue al tiempo máximo de horas para su reparación después de la falla (pero no puede producir nada mientras está descompuesta).
- La simulación trabajará únicamente con tres máquinas a la vez.
- En cualquier momento entre una simulación y otra se pueden cambiar las características de una o ambas máquinas para volver a ejecutar la simulación.
- En cualquier momento se puede cambiar un atributo en particular de la simulación o máquina (dejando el resto de los datos tal y como está) sin necesidad de volver a re-digitar toda la información.
- Todos los atributos relacionados a la máquina, ejecución, cambios o similares deben ser almacenados en la base de datos, así como el registro de sus creaciones, actualizaciones, y de ser necesario, eliminaciones.

Cada hora de la simulación se le consultará a cada máquina mediante un método para obtener los Productos Construidos En Esta Hora, que retorne la cantidad de productos construidos en esa hora, si la máquina se estropea en ese momento (en esa hora) se debe responder 0 y cambiar a estado de dañada. Se le deberán consultar tantas veces como horas se necesiten para repararse antes de que la máquina vuelva a producir. Este método simula una hora de producción cada vez que se invoca (y se debe invocar solamente una vez por hora simulada). El programa deberá contar con un menú desplegado en la vista principal (index). Este panel deberá solicitarle al usuario la opción que desea realizar o darle la opción de salir de la aplicación. El menú deberá permitir realizar las siguientes acciones, que pueden interpretarse en botones (que levantan modales o vistas):

- Obtener o modificar parámetros de configuración de la simulación (lleva a un menú con una opción por parámetro y otra opción para salir de ese sub-menú al menú anterior)
- Ingresar o modificar características de las máquinas (lleva a un menú con tres opciones, las dos primeras para cambiar los valores de la máquina 1 y la máquina 2 y la tercera opción para salir de este menú y regresar al anterior)
- Ejecutar la simulación. Mientras la simulación se ejecuta se deberá desplegar información importante (únicamente) cuando suceda alguna condición anómala, en otras palabras, cuando se dé una falla o se repare una máquina. Por ejemplo: En el mes: M, día: D, hora: H, de Producción se dañó la máquina Q – “Productos construidos por esta máquina a esta hora”: P - Ganancia en bruto a la hora del desperfecto: B - Ganancia real hasta ahora: R. En el mes: M2 día: D2 hora: H2 de producción fue reparada la máquina Q2 - Productos construidos por esta máquina a esta hora: P2 - Ganancia en bruto a la hora del desperfecto: B2 - Ganancia real hasta ahora: R2
- Ver resultados finales de la simulación más reciente (El sistema desplegará si sugiere comprar la máquina 1, la 2, ninguna o si para ese caso da igual cual equipo adquirir). Debe mostrar el detalle de información recolectada durante la simulación para ambas máquinas: Para la máquina 1: - Productos construidos por la máquina 1 durante la simulación: Cantidad de Productos - Ganancia en bruto durante la simulación: Ganancia en Bruto - Ganancia real durante la simulación: Ganancia Real. Para la máquina 2: - Productos construidos por la máquina 2 durante la simulación: Cantidad de Productos - Ganancia en bruto durante la simulación: Ganancia en Bruto - Ganancia real durante la simulación: Ganancia Real (ganancia en bruto menos costo de la máquina) Se recomienda “adquirir la máquina 1” o “2” o “ambas” o “ninguna”.
- Salir del programa (que cerrará la ventana del explorador de internet).

Se debe tener el servidor SQL (ya sea Cosmos DB o SQL Server) en Microsoft Azure, así como el hosting de la página web. Para esto, se debe crear un Grupo de Recursos llamado ProyectoFinal, donde se contenga la Base de Datos en Azure, con nombre proyectobd, el servidor del Plan de Aplicaciones llamado ServidorProyecto, y una Web App Service llamada “SimuladorVirtual-NombresApellidos”. La aplicación debe estar publicada, se puede incluir cualquier otro servicio de Microsoft Azure que se considere necesario (Application Insights, Azure Functions, Blob Storages, Key Vaults, etc...).

Entregables

- Aplicación Web de Visual Studio en .Net Core ASP MVC C#,
- Los Templates en JSON de los Servicios Creados en MS Azure.
- Este Archivo PDF con las Tablas de Calificaciones, con el nombre modificado.
- Un respaldo de la BD llena con los datos, en el formato de preferencia.
- Un Bloc de Notas (.txt) con el enlace (link) del proyecto publicado, y el link al repositorio público en GitHub dónde está sub-versionado.
- La fecha de entrega para el proyecto final es el Viernes 25 de Febrero del 2022 antes de las 11:59pm. La entrega debe hacerse de las siguientes maneras, todas simultáneamente, respetando el orden de prioridades, TODAS, antes de las 11:59pm del 25/02/22:
 1. A través del sistema Moodle de la plataforma universitaria en la correspondiente área de entregables para la última sesión.
 2. A través del correo electrónico institucional del profesor: ajimenez@ucenfotec.ac.cr, desde sus propios correos institucionales.
 - Crean una carpeta en su G-Drive institucional, llamada: "ProyectoFinal_NombresApellidos", pero en este que adjuntan un archivo comprimido con el proyecto (.ZIP o .RAR) y la documentación solicitada.
 - Adicionalmente en el cuerpo del correo, agregan también el enlace (link) al sitio web publicado y al GitHub dónde está el proyecto sub-versionado.
- El proyecto y sus documentos adicionales, deben ser sub-versionados en GitHub, y debe constar de al menos 4 commits, que no sean del mismo día. Pierden los puntos de este factor, si se presenta un solo commit, o commits múltiples del mismo día (asumiendo que no hay 4 repartidos en más de 4 días diferentes).
- Se recomienda que se empiece a trabajar desde hoy.
- Todo debe empacarse en un archivo compreso en formato principal de WinZip o WinRAR con el nombre ProyectoFinal_NombresApellidos. Y ser subido a la plataforma Moodle, en la correspondiente Sesión final, en la sección de entregables y enviados al correo institucional del profesor: ajimenez@ucenfotec.ac.cr, tal y cómo se indicó anteriormente.

Detalles de la Entrega:

- Cualquier proyecto que no compile o esté incompleto, la nota quedará a criterio del profesor.
- Este proyecto programado es únicamente individual y debe realizarse en el lenguaje de programación C#, utilizando el IDE de Visual Studio 2019 ó 2022. En ninguna circunstancia, se permitirán copias de trabajos.
- Si se presenta cualquier intento de fraude la calificación será de uno para todos los implicados.
- La fecha de entrega para este Proyecto Final es el **Viernes 25 de Febrero del 2022** antes de las 11:59pm. Ningún trabajo será recibido después de la fecha y hora establecidas. La entrega debe hacerse a través del correo electrónico mencionado anteriormente, pero primeramente en la plataforma Moodle, según se explicó, y en el que adjuntan un archivo comprimido en ZIP o RAR con el proyecto. Se recomienda que se empiece a trabajar desde hoy.
- En caso de detectar códigos iguales o muy parecidos entre proyectos las partes involucradas obtendrán la nota de 1 (uno).
- Se debe adjuntar este PDF en el archivo comprimido., según cómo se explicó en la sección anterior.
- Se advierte que la plataforma Moodle se cierra automáticamente, evitando subir más trabajos a la fecha y hora establecidos. Por eso deben darle prioridad a la plataforma.
- El día Viernes 25 de Febrero, 2022, será sólo para entrega y revisión de proyectos. No habrán clases presenciales, virtuales, sincrónicas, ni asincrónicas, ese día.

Criterios de Evaluación

Aplicación – 65 pts.

Utilización de la Estándares de Programación – 35 pts.

<i>Resumen</i>	<i>Puntos</i>
Aplicación	65
Utilización de Estándares	35
Total	100

Aplicación	Puntos
Definición de Clases, Manejo de la POO, Utilización MVC Web	10
Estructuras de Datos, Menús y Sub-Menús	10
Configuración de Parámetros de la Simulación	10
CRUD de configuración de las Características de las Máquinas	10
Ejecutar Simulación	10
Ver datos y resultados finales de la simulación y las máquinas	10
Sub-versionamiento y Commits	5
Total	65

Estándares de Programación Básicos	Puntos
Comentariado Estándar	8
Reutilización de Código (Código sin Repetir)	5
Definición y Nomenclatura de los nombres de: <i>clases, variables, métodos, etc...</i>	5
Declaraciones, Asignaciones y Excepciones	7
Estructura de las sentencias y ciclos	5
Código Funcional	5
Total	35

Documentación Oficial de MicroSoft sobre el Comentariado Oficial en C# ASP MVC .NET:
<https://docs.microsoft.com/es-mx/dotnet/csharp/language-reference/language-specification/documentation-comments>