UTN AVELLANEDA
Tecnicatura Universitaria en Programación

Integrador parte 1 Laboratorio II

Dominios:

- **Sistema único de boleto electrónico (SUBE).**
- **Bolsa de valores (Tipo cocos).**
- **Cuaderno de comunicaciones (para mandar informes a la familia)**
- **Administración de consorcio.**
- **Alquiler de volquetes.**

Dado el tema asignado, deberán cumplir con los siguientes requerimientos:

1. Realizar un análisis del dominio asignado. Deberán definir el ***alcance del sistema***, priorizando las funciones más importantes del mismo (el core del sistema).
2. Contar con un login de acceso a la aplicación y por lo menos dos tipos de usuarios (se recomienda un usuario final y un usuario administrador).
3. Administrar usuarios: El sistema deberá tener un rol súper usuario que podrá administrar a los usuarios finales (permitiendo dar de alta, baja y modificación de usuarios finales, así como otros objetos del sistema).
4. El sistema deberá tener un menú para navegar en la aplicación (recomendado).
5. Deberá usar la interfaz de múltiples documentos o MDI (recomendado).
6. La aplicación debe ser intuitiva y debe además estar distribuida en secciones de fácil navegación y acceso.
7. Debe haber al menos 4 formularios diferentes (4 vistas distintas).
8. La aplicación no podrá tener la apariencia por defecto (colores, iconos, nombres de forms, etc).
9. Se deberá separar la lógica de la aplicación en una biblioteca de clases.
10. Controlar todas las validaciones que se puedan dar.
11. El código deberá cumplir con los siguientes requisitos:
    a. Se deben respetar los cuatro pilares de la programación orientada a objetos: Encapsulamiento, Abstracción, Herencia y Polimorfismo.
    b. Contar con una jerarquía de clases con por lo menos 2 clases heredadas.
    c. Aplicar sobrecarga de constructores, operadores y métodos donde considere necesario.
    d. Usar distintos tipos de colecciones y por lo menos 2 (dos) enumerados.
    e. Usar métodos estáticos según corresponda.
    f. Serializar y deserializar archivos (JSON y XML).
    g. Investigación y uso de paquetes nuggets.
    h. Manejo de excepciones.

Notas:

1. Todos los métodos deben estar debidamente documentados
2. Aplicar principios DRY.
3. El programa no debe generar ningún tipo de error (ni de compilación, ni de ejecución).
4. Deben existir botones que completen los datos de un administrador o de un usuario final automáticamente para loguear la app (No loguearse automáticamente, sino completar la información en los textbox correspondientes para agilizar el ingreso de información).
5. Serán valorados a la hora de evaluar la creatividad que se aplique al programa en cuanto al diseño y usabilidad de la misma.
6. Se tendrá en cuenta el grado de dificultad del programa a la hora de la evaluación.

Algunos consejos para definir el alcance:

Identificar el propósito y los objetivos del programa:
Comenzar por comprender por qué están creando el programa y cuáles son los resultados deseados. ¿Cuál es el problema que intentan resolver o la necesidad que quieren satisfacer?
Podrían crear una lista inicial de funcionalidades del programa.

Analizar el público objetivo:
Determinar quiénes serán los usuarios finales del programa. ¿Para quiénes están diseñando el programa? ¿Cuáles son sus necesidades y expectativas?
Recuerden que deben existir dos perfiles. Un usuario final de la app, y un usuario administrador. Piensen qué necesidades de ambos perfiles debería cubrir el programa.

Investigar el dominio del programa:
Realizar una investigación sobre el tema o dominio que abordará el programa. Esto incluye entender las tendencias actuales, las mejores prácticas y las limitaciones técnicas.
A esta altura esta investigación ya debería estar parcialmente!!!

Crear una lista de funcionalidades y características:
Enumerar las funciones específicas que el programa deberá tener. Esto puede incluir funciones esenciales y opcionales. También es importante distinguir entre lo que es necesario y lo que es deseable.
Por ejemplo: me gustaría que mi programa consulte un servicio web que me actualice en la app constantemente la cotización del dólar. Me pregunto: ¿Esto es una necesidad, o es algo que me gustaría que haga mi programa?

Priorizar las funcionalidades:
Clasificar las funcionalidades y características en términos de importancia. Esto ayuda a enfocar los esfuerzos en lo más crucial y a dejar espacio para mejoras futuras.
Las funcionalidades más importantes serán las que evaluaremos ya que forman parte del core de la app.

Crear un diagrama de flujo o mapa conceptual:
Visualizar cómo interactúan las diferentes partes del programa y cómo se relacionarán las funcionalidades entre sí.
Ver para entender. Les recomendamos el uso de herramientas como drawio (web) o startuml (aplicación de escritorio) para diseñar clases, secuencias del programa, procesos, etc. No esperamos nada formal en este caso. Debe servirle a ustedes como apoyo, y forma de documentar lo que van a programar luego.
El uso de mockups para el diseño de las interfaces es muy útil. Les permitirá tener una noción de las vistas de la aplicación.

Ser flexible pero realista:
Reconocer que a lo largo del desarrollo pueden surgir cambios en el alcance. Es importante ser flexible pero también realista en cuanto a lo que se puede lograr.
Al principio pueden sentirse abrumados, la idea de esta experiencia es que puedan tomar decisiones sin la necesidad de que nosotros les digamos qué tienen que hacer, o cómo tienen que implementarlo.

Nosotros evaluaremos el proceso, y que puedan aplicar los temas que vamos viendo en la cursada. Las fechas de parcial son simplemente instancias en donde los profes pondremos una nota, pero el proceso de evaluación es constante. Incluye fechas de recuperatorios en cursada y mesas de finales de programación y laboratorio (es decir les podemos pedir que corrijan cosas hasta la última fecha en mesas de finales)

Metodología de trabajo:

- Podrán trabajar en grupos de 2 personas. Los participantes podrán dividirse tareas, pero tengan en cuenta que serán evaluados individualmente. Por lo tanto tienen que tener conocimiento íntegro del programa y de cómo fue implementado.
- Deberán crear un trello, para organizar el trabajo. El mismo lo tendrán que compartir con su tutor. Recuerden que vamos a reestructurar las tutorías por temática. Cada tarea a realizar tiene que estar identificada con su responsable. Trabajar con el formato de wall of work (tareas por realizar, tareas en proceso y tareas finalizadas)
- Crear un repositorio privado en Github (compartido con todos los profesores). Abriremos un formulario para que lo reporten. En el repositorio se deberá ver el trabajo de cada uno de los integrantes del grupo, es decir el avance del proyecto en github no puede depender de un solo miembro del grupo.
- Fechas de presentación: estarán publicadas en la hoja de ruta. Tengan en cuanta que por la cantidad de alumnos, solo podremos dedicar un tiempo limitado a cada grupo. Asegúrense de tener todo preparado y funcionando al momento de la presentación. Usaremos las clases de laboratorio y programación para evaluar.

UTN
AVELLANEDA
Tecnicatura Universitaria
en Programación