

Universidad Tecnológica
Nacional - FRBA

Diseño de Sistemas
2015

Profesor: Luciano
Straccia

TRABAJO PRÁCTICO INTEGRADOR

Cátedra de Diseño de Sistemas

Año 2015

Fecha de actualización: 21/10/2015

Tabla de contenido

| | |
|--|----|
| Introducción | 3 |
| El usuario | 3 |
| Las recetas | 4 |
| Datos de la aplicación | 6 |
| Entrega 1 | 8 |
| Entrega 2 | 9 |
| Entrega 3 | 10 |
| Entrega 4: Patrones de Diseño | 11 |
| Entrega 5: Diseño de base de datos | 12 |
| Entrega 6: Diseño de arquitectura | 13 |
| Entrega 7: Diseño de interfaz | 13 |
| Entrega 8: Pruebas | 14 |
| Anexo 1: Checklist de revisión del documento | 16 |

Introducción



Esta aplicación va a dar respuesta a la pregunta: “¿Qué podemos comer hoy?” Una problemática a la que se enfrentan diariamente casi todas las personas.

También va a dar respuesta *sobre qué conviene comer*, y además va a realizar recomendaciones diarias sobre *qué podría comer* acorde al perfil del usuario y otra información de interés general (estación del año,

temperatura, recetas que aún nunca seleccionó, etc.).

El sistema ofrecerá esta opción (sobre *qué comer*) siguiendo ciertos criterios ingresados por el usuario y siempre teniendo en cuenta las restricciones del perfil cargado.

Las opciones sobre *qué le conviene comer* dependerán de información que el sistema tiene del usuario tal como su contextura física, actividades que desarrolla, tipo de trabajo que realiza a diario, condiciones de salud básicas de la persona, entre otras.

Se pide desarrollar una aplicación WEB que cumpla dichos requerimientos.

El usuario

Un usuario del sistema podrá planificar sus cuatro comidas diarias (desayuno, almuerzo, merienda y cena) en base a un catálogo de recetas (ya existente) que puede ser extendido a gusto del usuario.

Podrá obtener sus estadísticas semanales y mensuales de los tipos de comidas y sus calorías (por ejemplo) por comida y consolidado.

Si el usuario no desea planificar sus comidas, puede usar el producto para simplemente consultar recetas nuevas y registrar las suyas y/o compartirlas en los grupos a los que pertenezca.

Las búsquedas siempre responden a las restricciones y preferencias del perfil cargado por el usuario.

Los usuarios del sistema tendrán la posibilidad de buscar una receta para cocinar en base a –entre otras cosas– su perfil nutricional. Este perfil tiene que ser previamente cargado y consta de la siguiente información de usuario: sexo, fecha de nacimiento, altura, complexión, preferencias alimenticias, condiciones preexistentes, dieta, rutina, entre otras. Anualmente el sistema solicita actualizar esta información.

También, el sistema automáticamente le ofrecerá al usuario las *recomendaciones del día*, en función del horario y preferencias del usuario y del grupo al que pertenece.

El sistema ofrecerá la posibilidad de administrar (crear, modificar, eliminar) grupos de usuarios, los que comparten sus intereses y preferencias (por ejemplo un grupo de usuario de entrenamiento). Los integrantes del grupo comparten todas las recetas, conocen quién las modifica y/o crea y pueden modificar una creada por otro integrante del grupo.

Si el usuario está interesado en una determinada receta para una comida deberá confirmarla incorporándola a la misma; además con esta acción la receta quedará incorporada a su historial. Confirmarla le permitirá al usuario conocer las recetas que cocinó en el pasado y también tener información para las estadísticas.

Otras recomendaciones que el sistema ofrece en función del histórico de recetas del usuario es ofrecerle sugerencias balanceadas en función de aquellas comidas que el usuario no suele comer. Es decir, las recetas no son sugeridas aleatoriamente sino que tendrán como objetivo balancear las diferentes recetas de la pirámide alimenticia.

El usuario tendrá accesible desde su home las últimas diez recetas aceptadas. Caso no haya tenido, mostrará aquellas que calificó y si tampoco dispone, las últimas que consultó. En caso de acceder la primera vez, mostrará las mejores diez calificadas.

Las recetas

Los usuarios podrán consultar las recetas que están cargadas en la base de conocimiento del producto. Al mismo tiempo le podrán hacer modificaciones o actualizaciones, pero dichos cambios solo serán visibles por el usuario y por el/los grupos a los que pertenece.



También tendrán la posibilidad de crear o inventar sus propias recetas. Estas recetas son únicamente accedidas por ese usuario y/o por el grupo, ya que no habrá un moderador que las apruebe y/o controle su contenido. Las mismas son responsabilidad del usuario. Solo el dueño de la receta puede actualizarla o borrarla (no los miembros del grupo).

Habrán recetas para ofrecer para las cuatro comidas diarias recomendables (desayuno, almuerzo, merienda y cena) aunque no es obligatorio que cumpla con las cuatro, podría hacer una o dos.

Cada una de las recetas debe ser clasificada a qué categoría de comida pertenece (cena o almuerzo) y podría pertenecer a todas las categorías. Se puede confirmar una única receta por comida o ninguna.

También se va a poder calificar las recetas, por lo tanto las sugerencias van a ser priorizadas u ordenadas implícitamente por este criterio. La calificación de las recetas precargadas es visible únicamente por todos los usuarios del grupo al que pertenece. Para la calificación se usa un criterio de 5 estrellas. Esta calificación es por gusto (1 no me gusta....5 me encanta) y es opcional.

Se van a poder buscar recetas de acuerdo a los diferentes criterios considerados: cantidad de calorías, ingrediente principal, ingredientes secundarios, temporada del año, dificultad de la preparación, eventos especiales, etc.

La idea también es aprovechar el dispositivo y la usabilidad, haciendo foco en un diseño centrado en el usuario. Se pide utilizar componentes del tipo: "drag and drop" de los ingredientes/condimentos para crear recetas.

Por último, se van a poder sacar estadísticas sobre el uso de la aplicación para seguir mejorándola. También el sistema debe ofrecer estadísticas y reportes de uso de los usuarios (por ejemplo: cantidad de recetas repetidas en el mes).

Datos de la aplicación

La aplicación tendrá disponible la siguiente información y datos:

- I. Ingredientes: Nombre, Porción, Calorías-Porción
- II. Condimento: Nombre, Tipo_condimento
- III. Dificultad_Receta: Fácil, Media, Difícil, Muy difícil
- IV. Temporada-Plato: Verano, Otoño, Invierno, Primavera (agregar eventos sociales según temporadas: Navidad, Pascua).
- V. Condiciones-Preexistentes: Diabetes, Hipertensión, Celíasis.
- VI. Complexión_Persona: Pequeña, Media y Grande.
- VII. Sexo: Femenino , Masculino
- VIII. Pirámide-Alimenticia: Nombre-Grupo, Descripción-Grupo, Contraindicaciones
- IX. Peso-Ideal-Hombres: altura, medida-tórax, medida-cintura, medida-cadera peso, peso-min, peso-max
- X. Peso-Ideal-Mujeres: altura, medida-tórax, medida-cintura, medida-cadera peso, peso-min, peso-max
- XI. Dieta: Normal, Ovolactovegetariano, Vegetariano, Vegano
- XII. Rutina-Ejercicio-Físico:
 - ☐ § Sedentaria con algo de ejercicio (-30 min.) LEVE
 - ☐ § Sedentaria con nada de ejercicio NADA
 - ☐ § Sedentaria con ejercicio (+30 min.) MEDIANO
 - ☐ § Activa con ejercicio adicional (+30 min.) INTESIVO
 - ☐ § Activa sin ejercicio adicional (+30 min.)

➤ Recetas

- Para cada receta: ALTA, MODIFICACIÓN y BAJA de los siguientes elementos:
 1. Ingredientes (lista de valores)
 2. Condimentos (lista de valores)
 3. Procedimiento (Explicación de la receta): 5 pasos – 5 fotos
 4. Dificultad de la receta (lista de valores: 1 a 5 estrellas)
 5. Temporada receta
 6. Calorías total de la receta
- Filtros de búsqueda de recetas según los criterios:
 - Ingrediente principal

- Temporada
 - Dificultad
 - Cantidad de calorías
 - Sector de la pirámide alimenticia
 - Calificación de los usuarios (del recetario).
- Catálogo de recetas: propias y online.
 - Selección de recetas del catálogo
- Perfil de usuario:
- a. Nombre
 - b. Sexo
 - c. Edad
 - d. Altura
 - e. Complexión (con ayuda, para que el usuario pueda determinarlo)
 - f. Dieta (lista de platos de opciones variadas para elegir)
 - g. Preferencias-Alimenticias (le gusta el pescado ó los mariscos, le gusta la carne roja, carne de pollo...)
 - h. Rutina
 - i. Condición-Preexistente: diabético, hipertenso, etc.



NOTA: Qué es lo que no contempla la aplicación:

ABM de datos maestros. Estos van a venir precargados en la base y no se van a poder modificar ni eliminar.

Entrega 1

Fecha de entrega: 06/05/2015

El objetivo de esta entrega es desarrollar el modelo de requerimientos funcionales y definir los requerimientos que impactará sobre las decisiones de diseño.

Para ello se solicita:

- Modelo de Casos de Uso para la aplicación
- Analizar cuáles atributos de calidad fueron considerados en el relevamiento que se encuentra en este documento. Aquellos atributos de calidad no considerados, definirlos en la especificación.
- Definir un repositorio de entregables. Informar al docente sobre la forma de acceso, los datos requeridos para su visualización y cualquier otra información necesaria.
- Explicar qué tecnología se utilizará para la implementación de algunos requerimientos. Indicar lenguaje de programación, bases de datos, framework y cualquier otra información que permita conocer la tecnología a utilizar.

Entre los criterios de evaluación se tendrá en cuenta:

- Consistencia entre el relevamiento y el modelo de casos de uso definido
- Correctitud en la identificación y análisis de atributos de calidad
- Correctitud en la definición de nuevos requerimientos asociados a los atributos de calidad
- Selección de un repositorio de entregables apropiado para las características del proyecto a desarrollar
- Completitud y claridad en las definiciones sobre tecnología

Entrega 2

Fecha de entrega: 24/06/2015

Se deberán especificar y diseñar los siguientes casos de uso:

- Cargar el perfil;
- Seleccionar recetas;
- Cargar recetas por perfil.

Para cada uno de ellos se solicita:

- Diagramas de UML que se consideran convenientes para el diseño
- Avances en la implementación de las funcionalidades mediante algún lenguaje de programación

Entre los criterios de evaluación se tendrá en cuenta:

- Consistencia entre el relevamiento y el análisis del dominio
- Consistencia entre los diferentes modelos/diagramas utilizados
- Correcta definición de los casos de prueba
- Concordancia entre la implementación realizada y las decisiones de diseño tomadas

Entrega 3

Fecha de entrega: 12/08/2015

Se deberá realizar el diseño (incluyendo el Diagrama de Clases y por lo menos otro diagrama UML) para las siguientes funcionalidades:

- Calificar las recetas;
- Administrar grupo del usuario;
- Estadísticas (por semana y por mes):
 - Según el sexo: tipos de receta más consultadas
 - Consultas según nivel de dificultad de la receta
 - Ranking de recetas más consultadas
- Reportes (según indicación del usuario):
 - Recetas consultadas en un período dado para un usuario
 - Recetas nuevas propuestas por los usuarios
 - Preferencias de recetas en un período determinado
 - Recetas según un rango de calorías a indicar por el usuario

Para cada uno de ellos se solicita:

- Diagramas de UML que se consideran convenientes para el diseño, incluyendo el Diagrama de Clases y por lo menos otro diagrama UML
- Implementación de las funcionalidades mediante algún lenguaje de programación
- Diseño de los casos de prueba

Entre los criterios de evaluación se tendrá en cuenta:

- Consistencia entre el relevamiento y el análisis del dominio
- Consistencia entre los diferentes modelos/diagramas utilizados
- Correcta implementación de las funcionalidades y cumplimiento de los requerimientos y diseño previstos
- Correctitud y completitud del diseño de casos de prueba

Entrega 4: Patrones de Diseño

Para el diseño realizado en la entrega anterior se debe:

- Aplicar patrones de diseño sobre los modelos, teniendo en cuenta mejorar la funcionalidad y minimizar el grado de acoplamiento entre los componentes. Se deberá justificar la aplicación de los patrones.
- Realizar la implementación de las funcionalidades mediante algún lenguaje de programación
- Registrar las decisiones de diseño (se sugiere el uso de la plantilla generada para tal fin)

Según la condición preexistente del usuario con respecto a si tiene: Diabetes, Hipertensión o Celíasis; la aplicación SOLO le propone determinadas recetas que son indicadas y aptas para esa condición del usuario y RECOMENDACIONES para la preparación de la misma. Para plantear dicha solución se propone aplicar el Patrón VISITOR.

Se pide rediseñar la solución utilizando las herramientas que crea convenientes para documentar la aplicación del patrón propuesto.

Entre los criterios de evaluación se tendrá en cuenta:

- Correcta aplicación de los patrones de diseño
- Consistencia entre los diferentes modelos/diagramas utilizados
- Concordancia entre la implementación realizada y las decisiones de diseño tomadas
- Cumplimiento de los ítems indicados en Anexo 1

Entrega 5: Diseño de base de datos

Para el diseño realizado en la entrega anterior se debe, en base al modelado realizado hasta el momento, definir un modelo de datos siguiendo el esquema relacional, justificando los componentes elegidos.

Se pide:

- 1) Confección del Diagrama de Entidad Relación (DER) en 3FN, definiendo:
 - a. Entidades,
 - b. sus atributos y dominios (tipos de datos) y
 - c. las relaciones entre entidades.
- 2) Mapeo de relaciones uno a uno, muchos a uno, uno a muchos y muchos a muchos.
- 3) Detalle de las decisiones de mapeo Objetos-Relacional que han sido tomadas
- 4) Implementación del modelo de datos en el motor de base de datos elegido
- 5) Implementación de las consultas para los siguientes requerimientos:
 - I. Para una Rutina-Ejercicio-Físico particular de usuario, las calorías máximas a consumir en una receta, por ejemplo para una rutina LEVE.
 - II. Mostrar todas las recetas con calificación 5 y de una determinada Temporada, por ejemplo verano.
 - III. Para un Dieta determinada mostrar las recetas disponibles, por ejemplo para un Vegano.
 - IV. Según Preferencia Alimenticias del usuario, mostrar 3 recetas posibles, como ser pescado o mariscos.
 - V. Para un Condimento en particular mostrar 5 recetas que lo contengan, por ejemplo con Romero.
 - VI. Para un determinado sexo y complexión determinada, mostrar las 10 recetas con una calificación determinada, por ejemplo sexo masculino complexión mediana y calificación de receta 3.
 - VII. Según Grupo Alimenticio de la Pirámide alimentaria mostrar 5 recetas, por ejemplo para el grupo de lácteos.

Entrega 6: Diseño de arquitectura

Se pide:

- 1) Definir la arquitectura que utilizará para implementar las Entregas anteriores. Justificar la decisión de la arquitectura elegida explicando por qué la implementación de dicha arquitectura favorece el diseño e implementación del producto.
- 2) Para el diseño de la arquitectura, realizar las siguientes Vistas de UML:
 - ✓ Diagrama de componentes
 - ✓ Diagrama de despliegue
- 3) Modificar el diseño y el código fuente según las decisiones de arquitectura

Se sugiere que conjuntamente con el diseño de la arquitectura se termine de implementar todas las funcionalidades sobre el producto y se diseñen los casos de prueba previstos para la Entrega 8.

Entrega 7: Diseño de interfaz

Se pide:

- 1) Asegurar que la aplicación sea Responsive y pueda ser utilizada desde dispositivos móviles.
- 2) Diseñar e implementar la interfaz de usuario considerando los atributos de calidad para la interfaz de usuario, principalmente:

I. Consistencia

El sistema debe comportarse siempre de la misma manera ante un mismo evento y las tareas similares deben poder realizarse siguiendo pasos similares (Cualidad del software; Passerini, Lombardi y Dodino)

II. Usabilidad

La usabilidad de una interfaz es una medida de la efectividad, eficiencia y satisfacción con la cual determinados usuarios pueden alcanzar determinados objetivos en un entorno particular con dicha interfaz (ISO 9241)

III. Robustez

Un programa es robusto si se comporta en forma razonable aún en circunstancias que no fueron anticipadas en la especificación de

requerimientos; por ejemplo cuando encuentra datos de entrada incorrectos o algún malfuncionamiento del hardware como rotura de disco. Un programa que genere un error no recuperable en tiempo de ejecución tan pronto como el usuario ingrese inadvertidamente un comando incorrecto no será robusto (Instituto de Computación, Universidad de la República, Uruguay)

IV. Navegabilidad

Una navegabilidad agradable es aquella que permite que las personas exploren el sitio de una manera sencilla, obvia e intuitiva con lo cual se produce una experiencia de visita satisfactoria que posibilite que los usuarios regresen y recomienden nuestra página web (vzert.com).

V. Manejo de errores de usuario y de sistema

Se asocia a la robustez.

Aseguren realizar el encapsulamiento de errores (try/catch o las sentencias apropiadas según el lenguaje) con mensajes apropiados para el usuario y no mostrar mensajes de error propios del servidor

3) Explicar de qué manera las interfaces de usuario desarrolladas permiten el cumplimiento de las características definidas en el punto anterior

Entrega 8: Pruebas

Se pide:

- Diseñar casos de prueba para las funcionalidades de estadísticas y reportes. Los casos de prueba deberán incluir: funcionalidad a probar, datos de prueba, resultados esperados, resultados obtenidos.
- Ejecutar los casos de prueba y reportar defectos (con su criticidad)
- Realizar el *testing* completo de la aplicación (no es necesario diseñar los casos de prueba en detalle para ser entregados, pero deberán asegurar que la aplicación funciona correctamente según lo previsto)

Entrega final

La entrega final incluye:

- Toda la documentación
- Código fuente y scripts de base de datos
- Producto funcionando

Entre los criterios de evaluación se tendrá en cuenta:

1. Concordancia entre la implementación realizada y las decisiones de diseño tomadas (y su respectivo modelado). Esto incluye las decisiones de clases, base de datos, arquitectura, interfaz de usuario.
2. Correcta ejecución de los requisitos solicitados en cada una de las entregas (aplicación funcionando correctamente)
3. Defensa grupal del trabajo realizado con preguntas individuales asociadas al producto construido y el proceso de desarrollo del TP

Para esta entrega final se sugiere disponibilizar el código fuente en un repositorio accesible por el equipo docente (a fines de poder realizar la evaluación del criterio 1) y además dejar disponible la aplicación ejecutándose en un servidor web (para la evaluación del criterio 2).

En caso que el grupo no pudiera dejar disponible la aplicación en un servidor web se deberá acordar con el equipo docente una fecha de revisión del funcionamiento de la aplicación fuera del horario de cursada de la asignatura.

Anexo 1: Checklist de revisión del documento

Se considera importante disponer de una serie de puntos a revisar antes de realizar la entrega, generando entregas consistentes y de mejor calidad. Estos puntos pueden ser referidos tanto a la comunicación del Diseño cómo a los aspectos de validar que el Diseño cumpla con los requisitos.

Los puntos a continuación son una propuesta a los que cada grupo puede agregar nuevos.

- Se evita la repetición innecesaria.
 - Está escrito desde el punto de vista del lector.
 - Se evita la ambigüedad y se usan notaciones.
 - El documento está organizado en partes estándares.
 - Se registran las decisiones de diseño.
 - La documentación presentada cumple con su propósito.
 - Se usaron herramientas de comunicación del diseño.
 - La redacción se hizo teniendo en cuenta los objetivos del proyecto.
 - La sintaxis del diagrama UML es correcta.
 - El diseño resuelve el problema planteado.
 - Los distintos diagramas son consistentes entre sí.
 - El diagrama de clases tiene atributos y métodos definidos que son relevantes para la entrega.
- El diseño está debidamente justificado.