

Trabajo Práctico 2015

¿Qué comemos?

Nos han solicitado el desarrollo de una aplicación que ayude a sus usuarios a organizar su dieta. Para ello, deberán suministrar varios datos sobre su salud y preferencias alimenticias, los cuales serán utilizados para recomendarle qué comer.



Deberemos construir la aplicación de punta a punta: desde el dominio hasta la interfaz gráfica y la persistencia. Los requerimientos detallados los iremos descubriendo y encarando a lo largo de las entregas.

[¿Qué comemos?](#)

[Entrega 0](#)

[Entrega 1](#)

[Explicación del dominio](#)

[Usuarios](#)

[Recetas](#)

[Requerimientos detallados](#)

[Punto 1](#)

[Punto 2](#)

[Punto 3](#)

[Punto 4](#)

[Punto 5](#)

[Hoja de evaluación del docente](#)

[Casos de prueba](#)

[Diseño general de la solución](#)

[Comunicación de la entrega](#)

[Entrega 2](#)

[Explicación del dominio](#)

[Sugerencias](#)

[Consultas](#)

[Requerimientos detallados](#)

[Punto 1](#)

[Punto 2](#)

[Punto 3](#)

[Punto 4](#)

[Punto 5](#)

[Punto 6](#)

[Hoja de evaluación del docente](#)

[Casos de prueba](#)

[Diseño general de la solución](#)

[Comunicación de la entrega](#)

[Entrega 3](#)

[Punto 1: Generar perfil de usuario](#)

[Modificaciones a las consultas](#)

[Punto 2: Nuevo origen de datos para las recetas](#)

[Punto 3: Monitoreo de consultas](#)

[Punto 4](#)

[Punto 5](#)

[Hoja de evaluación del docente](#)

[Casos de prueba](#)

[Diseño general de la solución](#)

[Comunicación de la entrega](#)

[Entrega 4](#)

[Punto 1](#)

[Punto 2](#)

[Cambios en el monitoreo de las consultas](#)

[Punto 3](#)

[Punto 4](#)

[Punto 5](#)

[Hoja de evaluación del docente](#)

[Diseño general de la solución](#)

[Comunicación de la entrega](#)

[Pre-Entrega 5: Prueba de concepto](#)

[MVC en Desktop](#)

[Layouts de pantallas](#)

[Login](#)

[Consulta de recetas](#)

[Hoja de evaluación del docente](#)

[Diseño de la interfaz de usuario](#)

[Comunicación de la entrega](#)

[Entrega 6: Persistencia a un medio relacional](#)

[Hoja de evaluación del docente](#)

[Diseño del mapeo entre el modelo de objetos y el relacional](#)

[Comunicación de la entrega](#)

[Entrega 7: persistencia a un medio no-relacional](#)

[Hoja de evaluación del docente](#)

[Diseño del esquema de persistencia](#)

[Comunicación de la entrega](#)

Entregas 1° Cuatrimestre

Entrega 0

El objetivo de esta primera entrega es configurar el entorno de trabajo de cada uno de los integrantes del grupo y tener un primer ciclo de desarrollo completo.

Se recomienda seguir las guías presentes en la página ddsutn.com.ar para configurar el entorno y crear el proyecto. Ver: <http://ddsutn.com.ar/material/software>.

Para esta primera entrega, se pide:

- Representar en código al “usuario”, que posee un peso y una altura
- Hacer que un usuario pueda calcular su índice de masa corporal (IMC), que se calcula según la siguiente fórmula: $\text{peso} / \text{estatura}^2$. Ejemplo: si pesa 83 kg. y mide 1,88 el IMC es: 23,48 aproximadamente.

Se pide que cada integrante del grupo implemente un caso de prueba unitario probando un caso de cálculo de IMC, con datos distintos (por ejemplo, cada uno podría usar su peso y altura). Cada integrante del grupo debe realizar al menos un commit al repositorio.

Para realizar los casos de prueba, dado que se trabaja con cálculos de punto flotante, se debe utilizar métodos que permitan comparar el resultado obtenido con el esperado utilizando una tolerancia. Es responsabilidad del grupo investigar en las guías de referencia de las tecnologías elegidas como realizar esto. La referencia que probablemente se quiera utilizar será la del framework de test elegido ([JUnit](#), [scalatest](#), etc).

Entrega 1

Dos conceptos fundamentales que maneja la aplicación son:

- los usuarios, de los que sabremos por ejemplo su información personal, médica y preferencias alimenticias
- las recetas, que los usuarios cargarán, y de las que nos interesará obtener información nutricional, entre otras cosas

El objetivo de esta entrega es entonces modelar a dichos usuarios y recetas.

Explicación del dominio

Usuarios

De los usuarios sabemos la siguiente información:

- Datos básicos como nombre, sexo, fecha de nacimiento
- Datos de la complexión: altura, peso
- Preferencias alimenticias: constituye un conjunto de palabras claves que indica si le gusta el pescado / los mariscos / carne / frutas etc.
- Comidas que le disgustan: un conjunto de palabras clave “Mariscos”, “Melón”, “Mondongo”.
- Condiciones preexistentes: si es diabético, hipertenso, celíaco o vegano. Las condiciones pueden combinarse y está pensado que puedan modificarse a futuro, ej: ovolactovegetariano, vegetariano, intolerante a la lactosa, etc.
- Rutina, que comprende alguno de estos valores:
 - 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.) INTENSIVO
 - Activa sin ejercicio adicional (+30 min.)

Además, los usuarios pueden agregar recetas al sistema; cada usuario conoce las recetas que agregó (ver más adelante).

Para que un usuario sea considerado válido, debe cumplirse que:

- tiene los siguientes campos obligatorios: nombre, peso, altura, fecha de nacimiento, rutina
- el nombre tiene más de 4 caracteres
- si el usuario es diabético, debe indicar el sexo
- si el usuario es hipertenso o diabético, debe indicar al menos una preferencia
- si el usuario es vegano no puede aparecer ninguna de estas palabras como preferencia: “pollo”, “carne”, “chivito”, “chori”

- la fecha de nacimiento debe ser una fecha anterior a la del día de hoy

Con estos datos además podemos:

- averiguar el índice de masa corporal o IMC de un usuario, implementado en entrega 0
- averiguar si un usuario sigue una rutina saludable. Esto se da si el IMC está en el rango de 18 a 30 y si no tiene condiciones preexistentes, o bien logra subsanar cada una de las condiciones preexistentes:
 - para los veganos, si le gusta las frutas
 - para los hipertensos, si tiene una rutina activa intensiva con ejercicio adicional
 - para los diabéticos, si tiene una rutina activa o no pesa más de 70 kgs.
 - para los celíacos no hace falta que cumpla nada.

Recetas

De las recetas sabemos la siguiente información:

- el nombre del plato
- una lista de ingredientes
- una lista de condimentos
- la explicación del proceso de preparación de la receta
- para qué condiciones preexistentes es inadecuado (se deduce, no es explícito)
- el total de calorías de la receta¹
- la dificultad de la preparación de la receta
- a qué temporada corresponde, inicialmente tendremos primavera, verano, otoño, invierno pero podríamos agregar Navidad, Pascua, Yom Kippur, Rosh Hashaná, Día de Gracias, etc.

Ejemplo: "Bifes a la criolla con papas y arvejas"

Inadecuado para: Veganos

Temporada: Todo el año

Total de calorías: 785

Dificultad de preparación: Media

Ingredientes:

Cuadril 3 kg	Papas 1,5 kg	Tomates triturados 1,5 kg
Morrón rojo 4	Morrón amarillo 4	Morrón verde 4

¹ El cálculo no es lineal, además de los ingredientes depende del tiempo y la forma de cocción

Cebollas 1 kg	Aceite de oliva 2 cdas	Huevos 20 unid
Tomate perita 1 lata	Puré de tomate 500 ml	Arvejas congeladas 500 grs

Lista de condimentos:

Ají molido C/N ²	Ajo picado 1 cabeza	Perejil picado C/N
Orégano C/N	Sal C/N	Pimienta C/N
Hojas de laurel C/N	Caldo de verdura o carne C/N	

Proceso de preparación:

1. Limpiar el morrón. Cortar en tiras a lo largo y reservar.
2. Pelar y cortar las papas en rodajas gruesitas, cortar la cebolla en juliana.
3. Rociar levemente con aceite de oliva a la paellera.
4. Pasar la carne por harina (para que se dore mejor) luego disponerla sobre la paella (previamente aceitada). Retirar la carne una vez sellada vuelta y vuelta.
5. En la misma paellera colocar una capa de cebolla cortada en juliana, pimientos en juliana, luego una capa de rodajas de papas, rodajas de tomates. Agregar la carne, condimentar por encima.
6. Una vez dispuestos todos los ingredientes en el disco con la sal, la pimienta, el orégano, el laurel y el ají molido, incorporar puré de tomate y caldo de verdura hasta cubrir la totalidad de la preparación.
7. Cuando las papas tomen color, indicará que la preparación está lista (la cocción total luego de agregar las papas lleva 20 minutos aproximadamente).
8. Agregar los huevos, chascándolos previamente por separado y luego volcarlos encima de los bifes. Sumar las arvejas y espolvorear con perejil picado toda la preparación.

Como se comentó anteriormente, un usuario puede agregar recetas, para lo cual se debe validar previamente lo siguiente:

- debe haber al menos un ingrediente
- el total de calorías deberá estar en el rango de 10 a 5000

Por otro lado, en realidad, las condiciones preexistentes para las que es inadecuada una receta deben ser deducidas a partir de las características de la misma:

- si la receta contiene sal o caldo no es recomendable para hipertensos

² C/N = Cantidad necesaria

- si la receta tiene más de 100 gr de azúcar como condimento no es recomendable para diabéticos³,
- si la receta contiene “pollo”, “carne”, “chivito” o “chori” en sus ingredientes no es recomendable para veganos.

Tener en cuenta que hay recetas todo el año que aplican a cualquier época.

Además, una receta puede tener subrecetas, para por ejemplo no tener que repetir la preparación del puré en las recetas “Bifes a la criolla con puré”, “Churrasco con puré”, “Pollo con puré”, etc. No hay límite al anidamiento de subrecetas.

Finalmente, las recetas pueden ser vistas o modificadas por un usuario si:

- Este usuario es quien la creó
- O bien es una receta pública: una receta que se sube en forma masiva por “sistema” (no las sube ningún usuario en particular)

Sin embargo, hay que tener especial cuidado con las recetas públicas al momento de modificarlas: si bien cualquier usuario puede hacer modificaciones o actualizaciones a una receta pública, dichos cambios sólo serán visibles por él.

Requerimientos detallados

Punto 1

- Indicar si un usuario es válido.

Punto 2

- averiguar el índice de masa corporal o IMC de un usuario
- averiguar si un usuario sigue una rutina saludable.

Punto 3

- Hacer que un usuario agregue una receta
- Conocer las condiciones preexistentes para las que una receta es inadecuada

Punto 4

- Saber si un usuario puede ver a una receta dada
- Saber si un usuario puede modificar una receta dada
- Modificar una receta dada, respetando la validación del item anterior
 - *Sugerencia: desarrollar casos de prueba para validar si un usuario X puede visualizar / modificar una receta R_1 privada de él, otra R_2 privada de otro usuario y una R_3 pública.*

³ Algunas recetas no indican la cantidad de azúcar (o establecen C/N, cantidad necesaria) en ese caso se considera despreciable (equivalente a 0)

Punto 5

- Poder construir una receta con sub-recetas.
 - *Sugerencia: generar casos de prueba que muestren cómo reutilizar la receta del puré dentro de cada una de las recetas principales.*

Hoja de evaluación del docente

Casos de prueba

- Calidad de los casos de prueba
- Casuística cubierta / Juego de datos preparado

Diseño general de la solución

- Manejo de errores / Separación de flujo principal y alternativo
- Buenas prácticas de diseño
 - evitar lógica duplicada,
 - mantener la simplicidad de la solución,
 - definición de la interfaz de los componentes que minimice su acoplamiento,
 - consistencia de la solución general
- Aplicación de conceptos de diseño/patrones dentro del contexto pedido

Comunicación de la entrega

- Trazabilidad de los requerimientos. Vinculación entre el análisis, diseño de alto nivel, diseño detallado, código y pruebas unitarias.
- Calidad general de la documentación presentada

Se hará una evaluación tanto grupal como individual de los conceptos de diseño implementados.

Entrega 2

Explicación del dominio

Queremos incorporar en esta entrega la posibilidad de crear grupos de usuarios, que comparten intereses y preferencias (por ejemplo un grupo de gente que entrena junta, o que está probando una dieta determinada). Cada grupo tiene un nombre: “Amigos del club”, “Ladies night”, etc. y un conjunto de palabras clave que referencia a preferencias alimenticias, que pueden no coincidir con las de cada uno de los usuarios.

La introducción de los grupos agrega una complejidad extra a la visibilidad de recetas: ahora, un usuario puede ver una receta también en el caso de que su creador comparta algún grupo con éste. Además de ver las recetas, si pertenezco al grupo puedo modificarlas.

Además se incorporan los siguientes requerimientos:

Sugerencias

El sistema puede sugerir una determinada receta a un usuario o grupo de usuarios. Esto ocurre cuando:

- en el caso de un usuario: la receta no debe contener ingredientes con palabras clave que le disgusten y además debe cumplir las condiciones preexistentes de su perfil. Ej: no ofrecer “paella” a alguien que le disgustan los mariscos, o una tortita negra a un diabético.
- en el caso de un grupo: la receta debe contener alguna palabra clave de la preferencia del grupo de usuarios y ser apropiada para todos los que conforman el grupo. Ej: si el grupo define que le gusta la pizza y la receta “Pizza 4 quesos” tiene sal, no debe haber ningún hipertenso en el grupo (tal como se lo definió en el punto 3 de la entrega 1)

Consultas

El sistema permite consultar por las recetas que un usuario puede ver (esto es, propias, públicas o compartidas a través del grupo). A la consulta se pueden aplicar varios cero o más filtros, como por ejemplo los siguientes (es deseable que sea fácil agregar nuevos filtros):

1. por exceso de calorías si el usuario tiene sobrepeso (se redefine para el grupo considerando el promedio). Sobrepeso = IMC, “exceso de calorías” > 500.
2. según las condiciones preexistentes del usuario o grupo de usuarios
3. porque no le gusta al usuario
4. por ser “caros” en su preparación, hay una lista de ingredientes caros: “lechón”, “lomo”, “salmón”, “alcaparras”, etc.

Debe ser posible combinar los filtros o intercambiarlos.

A una búsqueda se le puede asociar además un procesamiento posterior:

1. tomar los primeros 10,
2. o bien considerar únicamente los resultados pares,
3. ordenarlos por calorías, alfabéticamente o por cualquier otro criterio

Requerimientos detallados

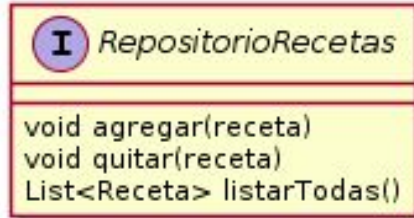
Punto 1

- Averiguar si una determinada receta puede sugerirse a un usuario o grupo de usuarios

Punto 2

- Conocer todas las recetas a las que un usuario tiene acceso.

Sugerencia: en algún punto será necesario conocer el conjunto de todas las recetas del sistema, o al menos, todas las recetas públicas. Para resolver esto puede ser de ayuda contar con un objeto con la siguiente interfaz o similar (podría tener más mensajes):



Punto 3

- Si el usuario está interesado en una determinada receta para una comida, el mismo deberá marcarla como favorita con el objetivo de dejarla en su historial para las búsquedas.

Punto 4

- Resolver los filtros y el manejo de resultados mediante Decorators.

Punto 5

- Resolver los filtros y el manejo de resultados mediante Strategies.

Punto 6

Hacer un análisis comparativo de ambas soluciones en una tabla donde se considere

- facilidad para agregar nuevos criterios de filtro
- posibilidad de eliminar filtros dinámicamente

- posibilidad de intercambiar el orden en que se aplican los filtros o el posterior manejo de resultados
- simplicidad
- cohesión o cuántos objetivos resuelve el componente desarrollado
- mantenibilidad

Hoja de evaluación del docente

Casos de prueba

- Calidad de los casos de prueba
- Casuística cubierta / Juego de datos preparado
- Evitar duplicaciones al generar casos de prueba

Diseño general de la solución

- Manejo de errores / Separación de flujo principal y alternativo
- Buenas prácticas de diseño
 - evitar lógica duplicada en las búsquedas,
 - mantener la simplicidad de la solución,
 - definición de la interfaz de los componentes que minimice su acoplamiento,
 - consistencia de la solución general
- Aplicación de conceptos de diseño/patrones dentro del contexto pedido

Comunicación de la entrega

- Calidad general de la documentación presentada
- Justificación de la utilización/desarrollo del componente que hace de repositorio de recetas

Se hará una evaluación tanto grupal como individual de los conceptos de diseño implementados.

Entrega 3

Punto 1: Generar perfil de usuario

Se pide que genere un componente que permita crear, eliminar, actualizar y consultar perfiles de usuario, respetando la siguiente interfaz

<i>RepoUsuarios</i>
<ul style="list-style-type: none">+ <i>add(Usuario)</i>+ <i>remove(Usuario)</i>+ <i>update(Usuario)</i>+ <i>get(Usuario)</i>: devuelve un usuario por su nombre+ <i>list(Usuario)</i>: devuelve una lista de usuarios cuyo nombre contenga el nombre del usuario ingresado. En caso de ingresar condiciones preexistentes, devuelve todos los usuarios que cumplan todas las condiciones preexistentes, ej: todos los diabéticos.

Puede utilizar como base `CollectionBasedHome<T>` del componente `uqbar-domain`. Esto permitirá desarrollar el caso de uso de negocio “Generar perfil”, que sigue este circuito

1. Un usuario nuevo genera su perfil y solicita incorporarse al sistema
2. El administrador de la aplicación ingresa al sistema y procesa las solicitudes pendientes, aprobándolas o rechazándolas. En el segundo caso debe indicar un motivo de rechazo.

Realice las modificaciones que crea necesarias.

Modificaciones a las consultas

Punto 2: Nuevo origen de datos para las recetas

Nos interesa incorporar un nuevo origen de datos para consultar las recetas, utilizando este servicio desarrollado en forma externa:

getRecetas: que acepta como valores un nombre, la dificultad, y una lista de palabras claves, devuelve como respuesta el siguiente formato JSON^{4 5}

- nombre de la receta
- lista de ingredientes
- el tiempo de preparación
- total de calorías
- la dificultad de la preparación de la receta:

⁴ JavaScript Object Notation es un formato de intercambio de datos más liviano que el XML. Para más información ver <http://es.wikipedia.org/wiki/JSON> y <http://www.json.org/>

⁵ Si les interesa buscar parsers para JSON, tienen <http://www.json.org/java/index.html>, <http://eclipsesource.com/blogs/2013/04/18/minimal-json-parser-for-java/>, <https://code.google.com/p/json-simple/wiki/DecodingExamples>, entre otros

- “D” difícil
- “M” dificultad media
- “F” fácil
- autor de la receta
- año en que se creó la receta

Las condiciones preexistentes las debe generar ud. utilizando como base el punto 3 de la entrega 1.

(la cátedra proveerá el componente y el juego de datos que devuelve)

Ud. debe diseñar con qué componentes debe interactuar para agregar esta funcionalidad al caso de uso Consultar recetas.

Punto 3: Monitoreo de consultas

Se pide que cada vez que se dispare una consulta se actualice la siguiente información:

- cantidad de consulta por hora del día. Ej: de 17 a 18 tuvimos 10 consultas, de 18 a 19 54 consultas, etc.
- cuáles son las recetas más consultadas. Ej: el “Pollo al Limón” apareció en 1742 consultas.
- discriminar las recetas más consultadas en base al sexo. Ej: los hombres consultaron más cómo hacer “Mollejas al verdeo”, las mujeres consultaron más cómo hacer “Matambre tiernizado de cerdo con papas noisette”.
- cuántos veganos hicieron consultas de recetas difíciles

por supuesto nos interesa poder agregar o eliminar monitores a futuro.

Debe utilizar observers para modelar este requerimiento.

Punto 4

Se pide que plantee un diseño alternativo al punto 3, sin utilizar observers.

Punto 5

Hacer un análisis comparativo de ambas soluciones en una tabla donde se considere

- facilidad para agregar o eliminar nuevos tipos de monitoreo (ej: total de consultas por usuario)
- simplicidad
- cohesión o cuántos objetivos a la vez resuelve el componente desarrollado
- mantenibilidad

Hoja de evaluación del docente

Casos de prueba

- Calidad de los casos de prueba
- Evitar duplicaciones al generar casos de prueba
- Utilización de objetos *stub/mock* para simplificar la unitariedad de las pruebas

Diseño general de la solución

- Manejo del asincronismo en el caso de uso Generar perfil
 - diferenciación entre los casos de uso de negocio y los de sistema.
Impacto en el diseño de la solución.
- Diseño de la interfaz saliente para la consulta de recetas
- Buenas prácticas de diseño
 - evitar lógica duplicada en las búsquedas,
 - mantener la simplicidad de la solución,
 - definición de la interfaz de los componentes que minimice su acoplamiento,
 - consistencia de la solución general
- Aplicación de conceptos de diseño/patrones para agregar funcionalidad a un objeto existente dentro del contexto pedido

Comunicación de la entrega

- Calidad general de la documentación presentada
- Análisis comparativo que muestre ventajas y desventajas de cada solución posible
- Calidad de la justificación presentada para elegir una opción u otra

Se hará una evaluación tanto grupal como individual de los conceptos de diseño implementados.

Entrega 4

Punto 1

Utilizar patrones creacionales para mejorar el fixture de los test que aplican a los perfiles de usuario.

Punto 2

Utilizar patrones creacionales para mejorar el fixture de los test que aplican a las recetas.

Cambios en el monitoreo de las consultas

El usuario nos pidió algunas mejoras en este punto:

1. queremos que las consultas de ciertos usuarios se envíen por mail. Ej: cada vez que el usuario “jcontardo” haga una consulta, debe enviarse los parámetros de búsqueda y la cantidad de resultados
2. queremos loguear las consultas que devuelvan más de 100 resultados⁶
3. algunos usuarios quieren marcar todas las recetas consultadas como favoritas. Para facilitar esta tarea, deben actualizar su perfil solicitando esta opción. En ese caso, al realizar la consulta debe marcar todas las recetas como favoritas. Si una receta ya es favorita la acción no tiene efecto.

Punto 3

Se pide implementar el Command Pattern para el requerimiento 1 (enviar mail determinadas consultas).

Punto 4

Se pide implementar el Command Pattern para el requerimiento 2 (loguear determinadas consultas).

Punto 5

Se pide implementar el Command Pattern para el requerimiento 3 (marcar como favoritas las recetas consultadas).

⁶ Utilizar para ello un framework de logging como [Log4J](#), [SLF4J](#), [Apache Commons Logging](#), etc.

Hoja de evaluación del docente

Diseño general de la solución

- Buenas prácticas de diseño
 - evitar lógica duplicada en las búsquedas,
 - mantener la simplicidad de la solución,
 - definición de la interfaz de los componentes que minimice su acoplamiento,
 - consistencia de la solución general
 - *diseño en el manejo de los errores*
- Aplicación de conceptos de diseño/patrones dentro del contexto pedido
 - representación del código ejecutable como abstracción de diseño

Comunicación de la entrega

- Calidad general de la documentación presentada
- Análisis comparativo que muestre ventajas y desventajas de cada solución posible
- Calidad de la justificación presentada para elegir una opción u otra

Se hará una evaluación tanto grupal como individual de los conceptos de diseño implementados.

Entregas - 2° Cuatrimestre

Pre-Entrega 5: Prueba de concepto

MVC en Desktop

Vista principal

El usuario tendrá accesible desde su home o pantalla principal las últimas diez recetas marcadas como favoritas. En caso de no tener, mostrará aquellas las últimas que consultó. En caso de acceder la primera vez, mostrará las diez recetas más consultadas.

Notas

- debe reutilizar el diseño de la vista cualquiera sea la información que esté mostrando
- debe simular el login para esta entrega

Modelo de Layout

Bienvenido a ¿Qué comemos?

Estas fueron tus últimas consultas

Nombre	Calorías	Dificultad	Temporada
Salmón con papas	550	Baja	Todo el año
Omelette de espárragos	400	Media	Verano
Bondiola a la cerveza negra con verduras	820	Difícil	Invierno

Ver

1. Se pide diferenciar las recetas públicas, las que creó el usuario y las que son de otro usuario que pertenece al mismo grupo: puede utilizar colores diferentes, o negrita/cursiva.
2. El botón Ver aparecerá inhabilitado, hasta que se seleccione una receta, entonces permitirá navegar hacia la vista de detalle de una receta (ver más abajo). En lugar del botón Ver puede utilizar un link en el nombre para navegar directamente sin necesidad de hacer dos clicks.
3. La leyenda que aparece antes de la grilla/tabla debe especificar el origen de los datos:

- En caso de aparecer las recetas favoritas debe decir “Estas son tus recetas favoritas”
- En caso de aparecer las últimas recetas consultadas debe decir “Estas fueron tus últimas consultas”
- Y si se está mostrando las recetas más consultadas, debe decir “Estas son las recetas top del momento”

Vista Detalle de una receta

Detalle de receta

Salmón con papas
550 calorías creado por vos

Dificultad
Media

Temporada
Todo el año

Ingredientes

Dosis	Ingrediente
800 g	Salmón rosado
6	Papas

Condimentos

Perejil
Ajo
Eneldo

☐ Favorita

Condiciones preexistentes

Veganos

Proceso de preparación

Ponés las papas en una olla con agua 30 minutos. El salmón se pone con ajo y perejil del lado de la piel en una plancha a fuego moderado durante ese tiempo. Antes de servir agregar el eneldo y listo.

Volver

En esta entrega se debe únicamente visualizar los datos de una receta.

MVC en Web Single Page Application

En el caso de trabajar con una tecnología SPA (Single Page Application) nuestra sugerencia es que divida la pantalla en dos subpaneles: del lado izquierdo estará la lista de recetas y del lado derecho el detalle de la receta seleccionada (el lado derecho ocupará un % mayor del total de la pantalla).

Servicios REST a implementar

Se debe generar un componente en el servidor que provea los siguientes servicios:

- Búsqueda de recetas principal,
 - recibe un perfil de usuario
 - devuelve la lista de recetas a mostrar en la página principal (según las definiciones anteriormente establecidas)
- Ver una receta
 - recibe el nombre de una receta
 - devuelve los datos completos de dicha receta

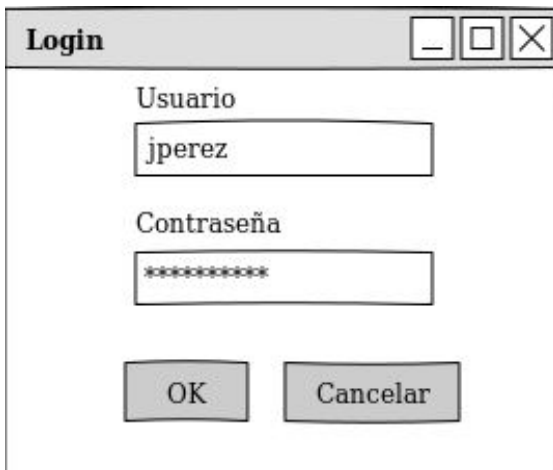
Entrega 5: UI

Debe definir una tecnología para implementar los siguientes casos de uso

- Pantalla de login
- Correcciones a la home principal
 - Refactor del detalle de una receta
- Ver perfil de usuario
- Consulta de recetas estableciendo criterios de filtro
- Monitoreo de consultas
 - Total de recetas consultadas en el día hora x hora

Layouts de pantallas

Login



A mockup of a login window titled "Login". It features a title bar with standard minimize, maximize, and close buttons. The main content area contains two input fields: "Usuario" with the text "jperez" and "Contraseña" with masked characters "xxxxxxxx". Below the fields are two buttons labeled "OK" and "Cancelar".

Se ingresa usuario y password, la contraseña no debe ser visible. Se valida contra el repositorio / home de usuarios construido en la entrega 3 (*RepoPerfiles*).

Pantalla principal

Además de la lista de recetas deben aparecer como menú de opciones:

1. Ver perfil de un usuario
2. Consulta de recetas
3. Monitoreo de consultas: Totales x usuario

y cuando se seleccione una receta debe llevarnos a la pantalla de Edición / Consulta de una receta (que anteriormente era siempre de sólo lectura)

Edición/Consulta de una receta

Detalle de receta

Salmón con papas

550 calorías creado por vos

Dificultad

Media ▼

Temporada

Todo el año ▼

Ingredientes

Dosis	Ingrediente
800 g	Salmón rosado
6	Papas

Agregar

Eliminar

Condimentos

Perejil

Ajo

Eneldo

Agregar

Eliminar

Condiciones preexistentes

Veganos

Actualizar

☐ Favorita

Proceso de preparación

Ponés las papas en una olla con agua 30 minutos. El salmón se pone con ajo y perejil del lado de la piel en una plancha a fuego moderado durante ese tiempo. Antes de servir agregar eneldo y listo.

Actualizar

Volver

- Si el usuario tiene permisos de edición, debe mostrarse el botón Actualizar y permitir la edición de los datos de la receta, incluyendo actualizar los condimentos, los ingredientes y las condiciones preexistentes. Si el usuario no tiene permisos para editar la receta, deben reemplazarse los campos editables por labels (etiquetas o textos) y no deben aparecer los botones Agregar/Eliminar Condimentos o Ingredientes ni el botón Actualizar. **Debe reutilizar la misma pantalla de la entrega 1 compartiendo los modos edición/sólo lectura.**
- En el caso de una edición, debe aplicar las validaciones de negocio de una receta diseñadas en la entrega 1.
- Para agregar un condimento, el usuario debe escribirlo en el texto que está arriba de la lista y presionar el botón Aceptar. Esta es la opción que se sugiere por simplicidad, cada grupo puede mejorar la idea hablando previamente con su tutor.

Página | 23 de 32

4. Para agregar un ingrediente, el usuario debe presionar el botón Agregar. Esto dispara una nueva pantalla que debe ingresar
- cantidad: numérico, entero, obligatorio
 - unidad de medida: opcional
 - ingrediente: obligatorio, se selecciona de una lista (sería bueno poder filtrar la lista a medida que se escribe un nombre)

Se adjunta una posible pantalla, el grupo deberá plantear los cambios que considere pertinentes:

Nuevo ingrediente

Cantidad <input type="text"/>	Ingrediente <input type="text" value="Me"/>
Unidad de medida <input type="text"/>	<div>Melaza Melón Mermelada</div>
<div>Aceptar Cancelar</div>	

5. Dado que las condiciones preexistentes las calcula el sistema (según lo establecido en el punto 3 de la entrega 1) no se cargan en forma manual sino que hay un botón para disparar dicha actualización.

Ver perfil de un usuario

Ver perfil de un usuario

Nombre

jcontardo

Sexo

Masculino

Fecha nacim.

15/06/1997

Complexión

Mediana

Altura

1,78

Peso

68,000

IMC

21,50

Preferencias alimenticias

Mariscos

Asado

Conejo

Comidas que le disgustan

Lentejas

Condiciones preexistentes

Rutina

NADA

Recetas favoritas

Paella a la valenciana

Cordero a las brasas con finas hierbas

Volver

1. Sólo se debe mostrar los datos del perfil de usuario, no se pide actualizarlos.
2. El campo IMC debe mostrarse
 - a. en verde si el valor es de 18 a 30
 - b. en rojo si el valor es superior a 30
 - c. en naranja si el valor es inferior a 18
 - d. redondear a dos decimales
3. La fecha debe mostrarse en formato "dd/MM/yyyy"
4. Debe respetarse el layout, cualquier modificación debe conversarlo con su tutor

Consulta de recetas

Nombre

Calorías

de a

Dificultad

 ▼

Temporada

 ▼

Que contenga ingrediente

☒ aplicar filtros del perfil de usuario

Buscar

Volver

Nombre	Calorías	Dificultad	Temporada
Salmón con papas	550	Baja	Todo el año
Omelette de espárragos	400	Media	Verano
Bondiola a la cerveza negra con verduras	820	Difícil	Invierno

Ver

1. **Debe reutilizar la pantalla principal** incorporando los campos de búsqueda que arriba se muestran.
 - a. Los campos de búsqueda son todos opcionales y tienen el siguiente comportamiento
 - b. nombre, busca que el nombre del plato contenga el valor ingresado
 - c. rango de calorías
 - i. rango desde: debe ser un número entero. Si se ingresa un valor x la receta debe tener más de x calorías
 - ii. rango hasta: debe ser un número entero. Si se ingresa un valor x la receta debe tener menos de x calorías.
 - d. dificultad y temporada: deben poder dejar la selección vacía, máximo un valor.
 - e. al ingresar un valor en el campo de texto ingrediente buscará que la receta contenga esa palabra entre los ingredientes

- f. si se marca el checkbox “aplicar filtros del perfil de usuario” sólo debe incluir las recetas que cumplan el perfil del usuario (puntos 4 y 5 de la entrega 2).
2. El usuario podrá ver / editar la receta al presionar en el link o el botón Ver.

Monitoreo de consultas: Ranking de recetas más consultadas

Esta pantalla debe mostrar las 10 recetas más consultadas

Ranking de recetas más consultadas[-] [x] [X]

Nombre	Calorías	Dificultad	Consultas
Salmón con papas	550	Baja	671
Pollo al limón	400	Media	577
Bondiola a la cerveza negra con verduras	820	Difícil	44

Volver

Notas:

- **Debe reutilizar la pantalla principal** incorporando la columna “consultas” que arriba se muestra
- Dependiendo la tecnología puede ser necesario incorporar un botón “Actualizar” para disparar nuevamente la consulta, o puede lograrse automáticamente.

Hoja de evaluación del docente

Diseño de la interfaz de usuario

- Atributos de calidad de la interfaz de usuario
 - Consistencia
 - Usabilidad: cómo guía al usuario a través del caso de uso
 - Robustez: cómo evita que el usuario cometa errores
 - Manejo de errores de usuario y de sistema
- Existencia de estrategias que permitan evitar la duplicación de código, con el objetivo de garantizar correctitud, minimizando la posibilidad de ocurrir errores durante la programación.
- Buenas prácticas de diseño
 - evitar lógica duplicada
 - mantener la simplicidad de la solución
 - definición de la interfaz de los componentes que minimice su acoplamiento
 - consistencia de la solución general
- Interacción entre componentes vista, controller, modelo, home
 - cohesión y acoplamiento entre dichos componentes
- Relación entre cada uno de los siguientes temas y su correspondiente implementación
 - layout
 - separación entre modelo y vista
 - manejo del estado conversacional de los datos que se cargan en la vista
 - tipo de binding utilizado
 - manejo de eventos
 - navegación
 - cómo es el flujo de una vista a otra
 - cómo se pasa información de una vista a otra

Comunicación de la entrega

- Calidad general de la documentación presentada

Entrega 6: Persistencia a un medio relacional

Partiendo de la implementación o especificación existente, ud. debe diseñar el mapeo entre el modelo de objetos de dominio (usuarios, recetas, etc.) y un esquema relacional.

- para cada uno de los siguientes ejemplos de impedance mismatch debe elegir un caso concreto y explicar qué decisión de diseño tomó
 - implementación relacional de la herencia en objetos
 - manejo de identidad
 - relaciones
 - uno a muchos
 - muchos a muchos
 - recursivas hacia la misma entidad(se pide explicar un ejemplo de cada tipo de relación)
 - tipos de dato (Ej: String sin especificar longitud máxima vs. CHAR(20), objeto Date y tipo primitivo Date, etc.)
- Implementar la solución utilizando un framework O/R M que soporte la especificación JPA⁷. Explicar
 - qué componentes del sistema se vieron afectados
 - cómo ayudó la arquitectura existente para minimizar el impacto de los cambios. Relacionarlo con el concepto acoplamiento.

Como resultado de esta implementación debe ser posible

1. ingresar al sistema
2. realizar cualquier operación que tenga efecto (colateral), por ejemplo
 - a. ver el perfil, marcar una receta como favorita
 - b. verificar la cantidad de veces que fue consultada una receta y consultarla 2/3 veces más
3. cerrar la aplicación
4. volver a abrir la aplicación y visualizar los cambios anteriormente generados
 - a. volver a consultar el perfil y ver que se agregó la receta antes marcada como favorita
 - b. ingresar al ranking de recetas más consultadas y verificar que aumentó la cantidad de consultas respecto al punto 2b.

⁷ o si lo desea puede diseñar un componente propio que realice el mapeo

Hoja de evaluación del docente

Diseño del mapeo entre el modelo de objetos y el relacional

- Justificación de la estrategia de mapeo definida para cada uno de los siguientes casos
 - herencia
 - manejo de identidad
 - relaciones
 - uno a muchos
 - muchos a muchos
 - recursivas
 - definición de tipos de dato
- Relación de la implementación o especificación del mapeo O/R y del componente de acceso a datos con las buenas prácticas de diseño
 - evitar lógica duplicada
 - mantener la simplicidad de la solución
 - definición de la interfaz de los componentes que minimice su acoplamiento

Comunicación de la entrega

- Justificación del impacto de los cambios pedidos en la arquitectura existente y su relación con el concepto acoplamiento
- Calidad general de la documentación presentada

Entrega 7: persistencia a un medio no-relacional

Se pide que diseñe e implemente un componente que garantice la persistencia de la aplicación a través de un esquema no relacional para manejar los siguientes casos de uso:

- Consulta de recetas favoritas
- Marcar/Desmarcar una receta como favorita
- Ver el detalle de las recetas

Como resultado de esta implementación debe ser posible

1. ingresar al sistema
2. marcar una receta como favorita
3. cerrar la aplicación
4. volver a abrir la aplicación y ver que aparece la receta como favorita para ese perfil de usuario

Hoja de evaluación del docenteDiseño del esquema de persistencia

- Definición de entidades y relaciones, incluyendo
 - atributos de las entidades y sus dominios / tipos de datos
 - relaciones uno a muchos y muchos a muchos
- Buenas prácticas del modelo de datos definido
 - Grado de normalización / redundancia
 - Criterio de particionamiento utilizado (si corresponde)
- Relación de la implementación / especificación del esquema de persistencia y del componente de acceso a datos con las buenas prácticas de diseño
 - evitar lógica duplicada
 - mantener la simplicidad de la solución
 - definición de la interfaz de los componentes que minimice su acoplamiento
- Flexibilidad para configurar el esquema de persistencia en forma dinámica

Comunicación de la entrega

- Justificación del impacto de los cambios pedidos en la arquitectura existente y su relación con el concepto acoplamiento
- Calidad general de la documentación presentada
 - Comparación contra el modelo de persistencia relacional, analizando debilidades y fortalezas