



Nombre de la materia

Diseño de sistemas

Nombre de la Licenciatura

Ingeniería en Computación

Nombre del alumno

González Malvaez Luis Omar

Ibarra Romero Antonio

Muñiz Reyes Alejandro Ismael

Olvera Poumian Roger Shayr

Páez Moedano Alan Eduardo

Rodríguez García Juan

Rojas Santillán Montserrat

Matrícula

1728978

1728988

1729002

1729005

1729009

1729016

1729023

Nombre de la Tarea

Seguimiento de proyecto - Documentación

Unidad # / Actividad

2/-

Nombre del Tutor

XX

Nombre del Profesor

L.I.A. Yeni Elizabeth Puente García

Fecha

08 de noviembre de 2021



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



Parte 1

Arquitectura elegida: “Por capas”

Se eligió la arquitectura por capas para nuestro proyecto porque hace más fácil de entender y organizar el desarrollo de nuestro sistema ya que nos proporciona una distribución de las tareas y responsabilidades en una forma de mayor o menor importancia o relevancia dentro de la misma, además de separar de forma segura los roles que desempeñara cada parte del sistema teniendo una secuencia de interfaces muy claras entre las capas. Complementado con lo anterior las tres capas que maneja esta arquitectura nos ayuda a decidir cómo organizar el diseño de cada capa y por qué elementos del sistema estará conformada cada capa e indicar con claridad que pueden hacer.

Permitiendo medios para aislar las capas del proyecto y poder realizar o aplicar cambios en el funcionamiento interno de alguna de estas sin la necesidad de alterar o recrear alguna de las otras capas. La realización de pruebas es favorable porque da la posibilidad de cambiar capas completas siempre que se mantenga la interfaz y debido a las características de la arquitectura es posible la portabilidad sin grandes complicaciones, esto gracias a que se puede probar cada capa de forma individual sin necesidad de alterar a las demás, a su vez esto facilita el mantenimiento.

Con esta arquitectura tendremos cierto nivel de seguridad para nuestros clientes ya que nos permite aislar las capas en subredes haciéndolo más seguro y resistente a cualquier tipo de ataque.

Patrón de diseño elegido: Abstract Factory

En el patrón Abstract Factory, permite mediante una interfaz responsable la creación de una fábrica de objetos relacionados sin la necesidad de especificar explícitamente sus clases. Cada fábrica generada puede entregar los artículos según el modelo de fábrica. Los modelos abstractos de fábricas funcionan alrededor de una súper fábrica que crea otras fábricas. Primero se sigue la declaración de interfaces forma explícita para cada

producto (o en nuestro caso, tipo de café) para después hacer todas las variantes disponibles. Aunque por el momento contamos solo con tres categorías y un solo producto (café) esto abre una gran posibilidad de opciones para ser adaptado no solo en cafeterías, sino en muchos otros ambientes de servicio a futuro.

Esta elección de patrón, además, nos brinda más flexibilidad para el aislamiento de clases en concreto, esto será bastante útil ya que podremos trabajarlo de una forma más sencilla para ambas partes además de ser un poco más simple y fácil de entender adaptándose a nuestro proyecto sin ningún inconveniente.



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



Modelo: Vista Controlador

El MVC separa los datos de la aplicación, la interfaz de usuario, y la lógica de control. El Modelo que contiene una representación de los datos del sistema, su lógica de negocio y sus mecanismos de persistencia. La interfaz de usuario con la información que vera el cliente y sus mecanismos de interacción. El Controlador que es el intermediario entre el Modelo y la Vista el cual gestiona el flujo de información entre estos dos junto con las transformaciones para adaptar los datos que sean necesarios según las solicitudes.

Parte 2

Metodología elegida: RAD

Se ha elegido esta metodología ya que consideramos que es la más adecuada al desarrollo de nuestro sistema; hicimos una recopilación de requisitos mediante cuestionarios con el barista, hasta el momento llevamos trabajando un prototipo que se ha puesto a prueba con algunos usuarios, reutiliza la maquina virtual de las aplicaciones en Java, ofrece la posibilidad de desarrollar o modificar el sistema de manera más rápida y eficiente en dado casos de requerirse y además hacemos uso de herramientas de desarrollo rápido como lo es el GUI de Apache NetBeans, donde se está desarrollando el sistema en su forma visual

RAD encaja muy bien en nuestro trabajo actual y seguimos aplicando algunos de sus fundamentos como lo es el programa de ritmo rígido con el cual se posponen algunas mejoras para la siguiente versión, dedicando mayor tiempo al arreglo de características incompletas o disfuncionales del sistema.

Fases de la metodología RAD

Consta de cuatro fases. La primera fase es llamada "Requisitos de planeación", donde los desarrolladores, clientes y miembros del equipo se comunican entre si para determinar las metas y las expectativas para el proyecto.

La segunda fase es llamada "Diseño de usuario", con la cual se entra de lleno con el desarrollo. Los clientes trabajan de la mano con los programadores para asegurarse de que de que los requerimientos sean cumplidos casi como si se tratara de software personalizable donde los usuarios pueden probar los prototipos en cada etapa para así asegurarse de que cumpla con las expectativas.

La tercera fase es "Construcción rápida", donde se toman los prototipos y sistemas en beta de la fase de diseño, para que los programadores puedan *construir* el modelo funcional final de manera mas rápida a comparación de una metodología tradicional.

La cuarta fase es llamada "Limpieza" o "Cutover", que no es más que la fase de implementación, donde el producto finalizado se lanza. Incluye todos los datos de conversión, pruebas, y de cambios al sistema nuevo,



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



así como el entrenamiento del usuario final. Todos los cambios finales son hechos mientras los programadores y clientes continúan buscando bugs en el sistema.

Ciclo de vida del sistema de información:

-Planificación: Durante esta etapa abordamos la problemática que se presentaba en diversas cafeterías de alta demanda; la atención a clientes inexpertos de manera más personal, y comenzamos el desarrollo de nuestra solución propuesta la cual fue un sistema experto que retomara los aspectos más básicos e importantes que se toman en cuenta al momento de elegir un café. Con esto el sistema experto proporcionaría a los clientes un café acorde a su preferencia y, al mismo tiempo, libraría de carga laboral a los baristas humanos en turno.

De primera instancia el alcance de nuestro sistema experto se limitará solo a los cafés, pero conforme pase el tiempo podrían agregarse nuevas características conforme se vayan requiriendo.

-Análisis: El sistema experto deberá de ser capaz de permitir a los usuarios elegir un café que se asemeje a sus gustos personales dependiendo de tres factores: Temperatura, Sabor e Intensidad. Dependiendo de estos tres aspectos debe poder recomendar un café al cliente que satisfaga sus necesidades

-Diseño: Durante esta etapa se optó por diseñar el sistema experto en Java, para ser mas concretos en un GUI que funciona con las elecciones del usuario, el sistema guarda y analiza la elección del usuario para determinar su café ideal. El sistema experto cuenta con un GUI intuitivo que puede ser usado tanto en un equipo con ratón y teclado tanto como en pantallas táctiles dotándolo de una amplia gama de aplicabilidad en diferentes dispositivos. Al ser creado en Java los equipos requieren menos especificaciones técnicas necesarias u obligatorias pues gracias a la maquina virtual de Java el sistema puede ser ejecutado en prácticamente cualquier dispositivo, permitiendo escatimar gastos en equipo necesario o especializado para poder ejecutar el sistema.

-Integración (Pruebas): Hasta el momento se ha puesto en marcha una versión alfa, que nos ha permitido llevar a cabo una serie de pruebas en un ambiente controlado. El sistema aún se encuentra en desarrollo y las mejoras tanto de diseño como funcionalidad están presentes en cada ciclo de integración que se da para pulir detalles que se lleguen a presentar. Cuando el sistema cumpla los estándares deseados podrá pasar a la fase de implementación en distintas cafeterías y conforme pase el tiempo y los requisitos se amplíen o se desean agregar nuevas características pasaremos a la etapa de mantenimiento

-Implementación: //

-Mantenimiento: Cada cierto tiempo se planificarán revisiones semanales, mensuales, trimestrales, etc, según se requiera para poder monitorear y analizar el comportamiento, detectando así errores y/o dar paso a posibles optimizaciones dando así un mantenimiento y actualización adecuados, minimizando de esta manera el margen de error lo más posible e incrementando la comodidad del usuario mediante la actualización de las interfaces haciéndolas más amigables para estos.



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



Parte 3

Diseño del sistema

Para el diseño de la Potrobarra se optó por la arquitectura por capas y de esta manera pudimos basar nuestro sistema experto en diferentes capas que a su vez funcionan como estados de elección. Conforme el sistema va registrando las respuestas del usuario se van guardando dichas elecciones y al final el sistema puede determinar el café ideal basándonos en los estados (o respuestas) registradas.

A continuación, se mostrará la interfaz de usuario, seguida de una explicación de que hace cada parte:

Esta es la pantalla de inicio, es fácil de usar por que el cliente solo debe de escoger entre sí o no:





1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



En esta pantalla se seleccionará la temperatura ideal de la bebida del cliente, en ambas opciones se repiten algunos tipos de café ya que las opciones de preparación son muy extensas:



En esta pantalla se escoge si la bebida será dulce o amarga. Algunos clientes podrían tener diabetes, es por esa razón que esta pregunta es importante:





1 / 1: Proyecto / Seguimiento del proyecto


Diseño de sistemas




Se escoge si la bebida será intensa o suave. Algunos clientes podrían no soportar algunas texturas de la bebida:

¿Intenso o Suave

Intenso



Suave



Después se confirmará la bebida recomendada, el nombre del cliente y su pedido con un ticket.

Nuestra recomendación es:



Café: Moccachino Frio Ultra-expresso

Nombre del cliente: Juan Camaney

No. De Ticket: 69420666

Confirmar



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



Bibliografía

- Castro, L. M. (2014). Arquitectura Software [Diapositivas]. madsgroup.
<https://upload.wikimedia.org/wikipedia/gl/b/be/AS1-arquitecturas-no-distribuidas.pdf>
- Costanzo, M. (2020). Platzi: Cursos online profesionales de tecnología. Platzi.
<https://platzi.com/tutoriales/1248-pro-arquitectura/5439-patron-arquitectonico-de-capas-layers/>
- ECONSTOR. (2015, octubre). ARQUITECTURA DE SOFTWARE ACADÉMICA PARA LA COMPRENSIÓN DEL DESARROLLO DE SOFTWARE EN CAPAS (N.o 574).
<https://www.econstor.eu/bitstream/10419/130825/1/837816424.pdf>
- García Mariscal, A. B. (2015). MODELO DE PROGRAMACION WEB Y BASES DE DATOS (1.a ed.). Editorial Elearning.
- Yáñez, J. M. C., & González, J. Á. T. (2005). Sistemas de Información Medioambiental. Netbiblo.
- Refactoring.Guru.. (2021). Abstract Factory. 18 de septiembre de 2021, de Refactoring.Guru. Sitio web: <https://refactoring.guru/es/design-patterns/abstract-factory>
- Programación y más.. (24 de febrero de 2014). Patrón de Diseño Abstract Factory. 18 de septiembre de 2021, de Programación y más. Sitio web: <https://programacionymas.com/series/patrones-de-diseno-en-java/patron-de-diseno-abstract-factory>



1 / 1: Proyecto / Seguimiento del proyecto

Diseño de sistemas



- Santander Universidades. (21 de diciembre de 2020). Metodologías de desarrollo de software: ¿qué son? 18 de septiembre de 2021, de Santander Becas Sitio web: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- 19/Mar/2013 2001-2021 © Línea de Código. Sitio web: <https://lineadecodigo.com/patrones/patron-abstract-factory/>
- Arquitectura de capas. (s. f.). Arquitectura de Capas. Recuperado 19 de septiembre de 2021, de <https://somospnt.com/blog/118-arquitectura-de-capas>