

Universidad del Valle de Guatemala
#17276

Programación Orientada a Objetos
#17032

Catedrática: Lynette García Pérez
Sección 21

Mayra Silva

Odalís Reyes

Ivan Maldonado #17211

PROYECTO ENTREGA FINAL

El nombre de la base de datos es Proyecto y el nombre de la colección es "iOrder".

I. RETROSPECTIVA

A. Descripción del problema

Un problema que afecta el día a día de los guatemaltecos, es el tráfico; un fenómeno que va en aumento con el paso de los años. Esto se debe al crecimiento de la densidad poblacional, lo cual conlleva al aumento de demanda en cuanto al transporte y en vista de la inseguridad que existe en el país, la falta de acceso a transporte colectivo seguro, la mejor alternativa para transporte es un carro. Sin embargo, el aumento de la demanda de carros ocasiona un congestionamiento vehicular en horas pico y también en horas fuera de ese rango.

B. Necesidades detectadas

Entre las personas entrevistadas se detecta que en su mayoría, surgen necesidades de alimentación, fisiológicas como de ir al baño y mentales como ansiedad y estrés.

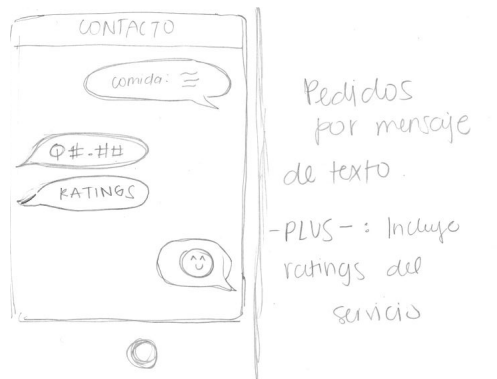
C. Oportunidades por cada una de las necesidades detectadas

- Alimentación: un programa que permita al usuario tener una forma de conseguir alimento de una forma accesible desde el tráfico, segura y que sea del agrado del usuario.
- Necesidades fisiológicas: una interfaz que facilite al usuario acudir a un baño, de tal forma que no pierda su lugar en el tráfico, que sea segura y limpia.
- Necesidades mentales: programas de relajación y meditación, música adecuada al ambiente, juegos que permitan quitar el estrés, programas que permitan mantener al usuario entretenido fuera del estrés.

D. Descripción de la fase de ideación

La generación de ideas se llevó a cabo mediante la realización de un proceso de lluvia de ideas, el cuál consistió en realizar tres sketches por persona que reflejaran cada necesidad propuesta dentro de la problemática del proyecto. Estos se crearon mediante un dibujo hecho por cada integrante del grupo para discutir las ideas propuestas brevemente de cada uno. Al final se obtuvo un total de 12 sketches.

Sketch 1



Sketch 2



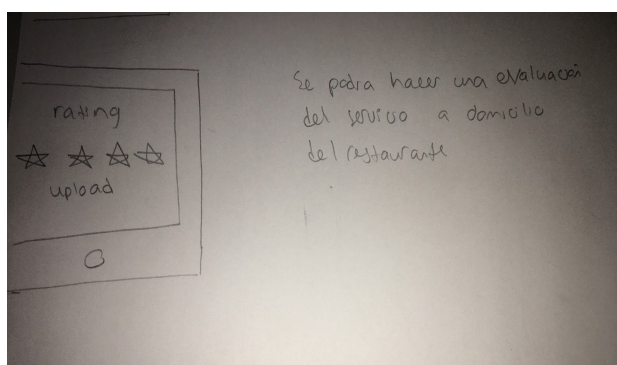
Sketch 3



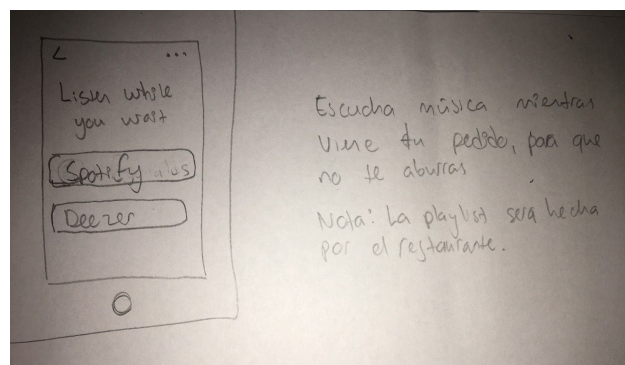
Sketch 4



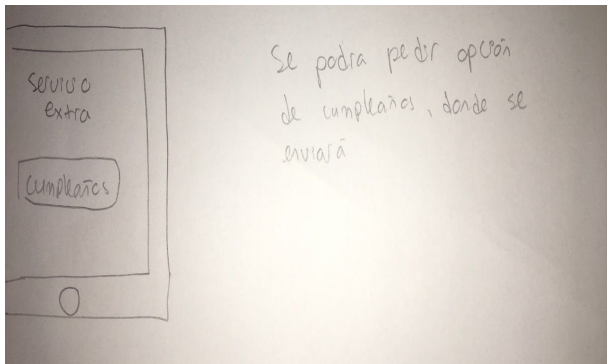
Sketch 5



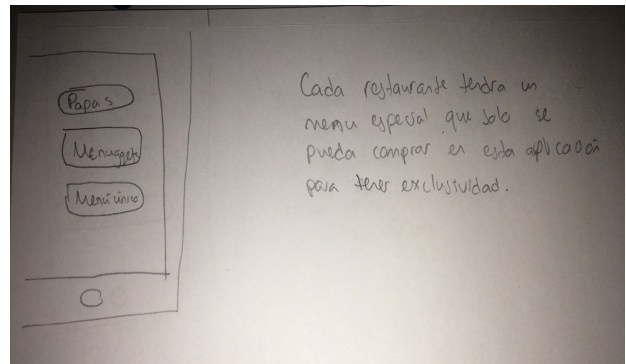
Sketch 6



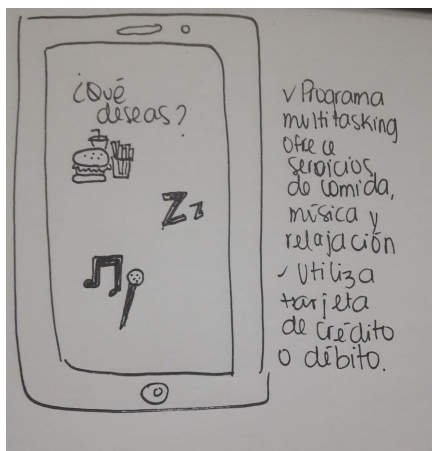
Sketch 7



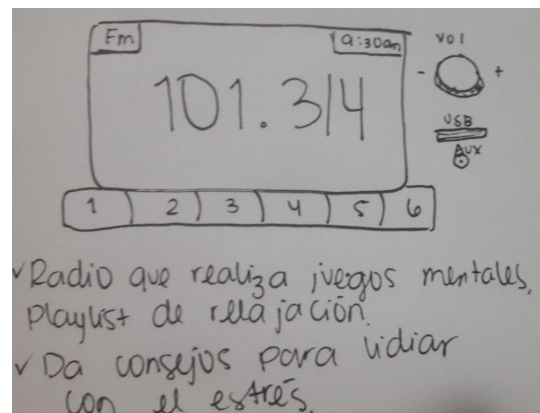
Sketch 8



Sketch 9



Sketch 10



Sketch 11



Sketch 12



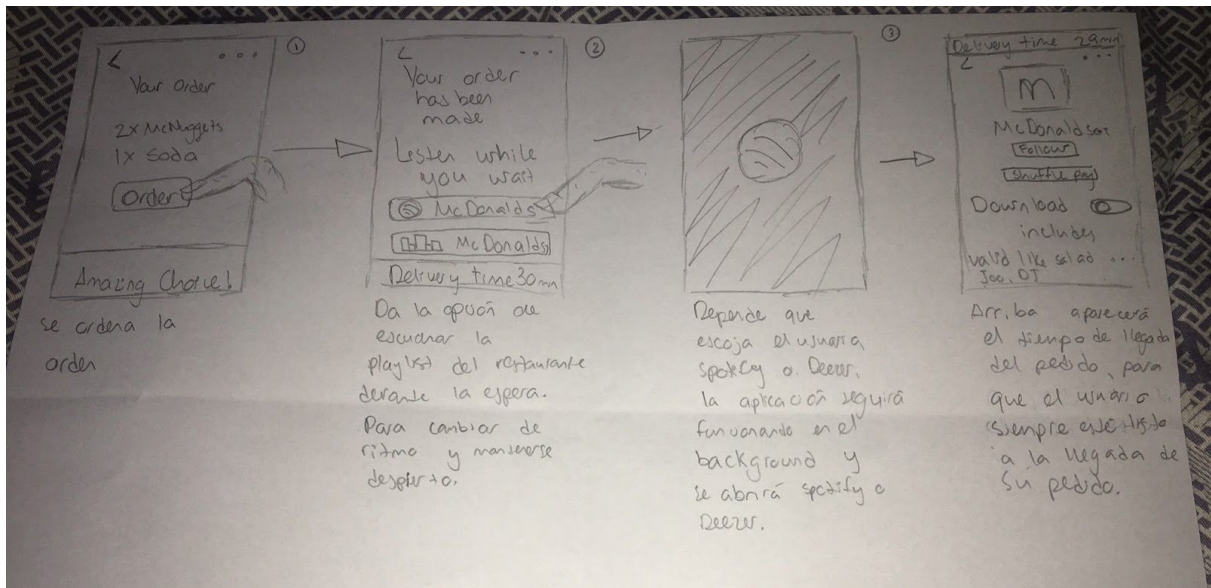
Por último, se escogieron las tres mejores ideas para realizar los storyboards y realizar una breve descripción de cada una.

E. Descripción de las ideas más votadas

- **(Combinación de Sketch 6 y 8):** El usuario podrá distraerse y mantenerse despierto utilizando la playlist especial del restaurante a consumir después de haber realizado su orden. Por otro lado, se mantendrá informado del tiempo de llegada de su alimento seleccionado.
- **(Sketch 1):** Mientras la persona se encuentra en el tráfico, podrá enviar un mensaje de texto a un número en específico donde le proporcionará diferentes números de extensiones para realizar un pedido de comida (pizza hut, pollo campero, etc). Luego, ingresa su orden y le aparecerá que su orden está registrada, junto con el monto de dinero de su orden. Después, le aparecerá una encuesta para mejorar la calidad de servicio y después de un intervalo pequeño de tiempo, su comida será entregada mientras la persona esté en el tráfico.
- **(Combinación de Sketch 9 y 12):** El usuario podrá ingresar sus datos y su número de tarjeta (con el propósito de que la app guarde sus datos para futuras ocasiones). Luego, cuando este en el tráfico, el usuario podrá ingresar a la aplicación y elegir una necesidad: comida o relajación. Si el usuario escoge comida, le preguntará que tipo de orden desea. Luego, lo conectará a una app musical para reproducir una playlist para relajarse o para desestresarse.

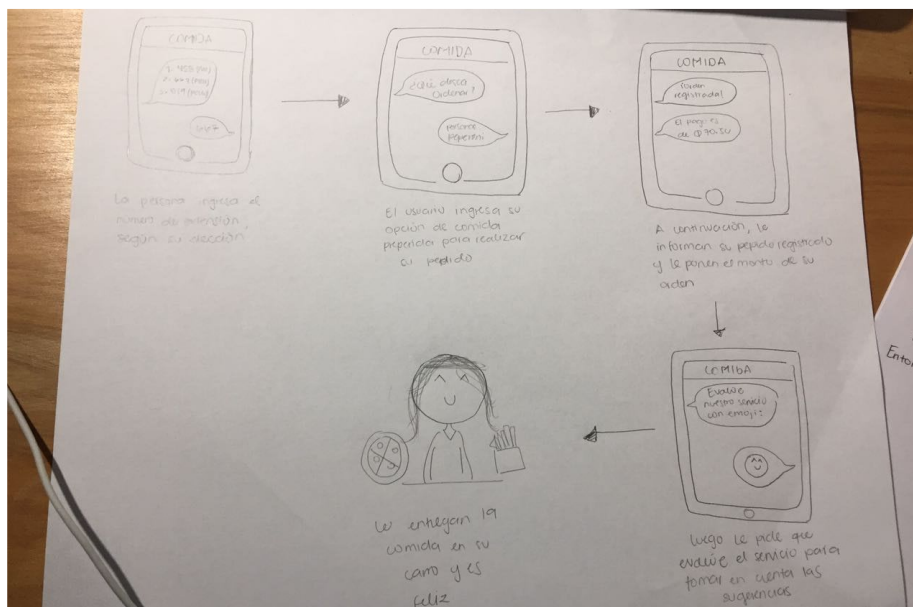
F. Descripción breve de prototipos elaborados

Storyboard #1



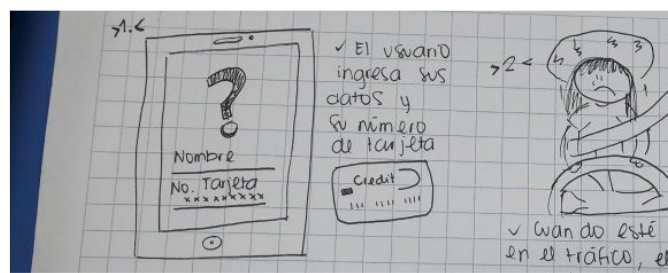
- **Comentarios:** Esto se hizo en base a que los usuarios opinaban que una buena radio los mantendría despiertos y creemos que variar un poco de la música que el usuario normalmente usa, sería una buena forma de mantenerlo activo y despierto.

Storyboard #2



- **Comentarios:** Consideramos que es buena idea, sin embargo es peligroso estar enviando mensajes de texto mientras la persona conduce. La idea está bien pensada, pues complace la necesidad general del conductor: el hambre mientras está en el tráfico.

Storyboard #3



- *Comentarios:* Buena idea, es una propuesta que satisface ambas necesidades principales (las cuales es la relajación y la alimentación). El usuario podría ser capaz de cumplir y estar relajado mientras conduce en el tráfico con esta opción.

II. IDENTIFICACIÓN DE REQUISITOS FUNCIONALES

- El programa debe de ser capaz de darle la opción que el usuario cree una cuenta en la aplicación, de tal manera que los datos que va ingresando se van guardado. Este tipo de información solo será visible para el mismo y para la aplicación, de tal manera que los restaurantes también tengan acceso a sus datos para completar las ordenes.
- Deberá ingresar su número de tarjeta de crédito (esto sólo será visible para el usuario y para el sistema para poder cobrar el monto de dinero). Es decir, aparecerá encriptada al igual que la contraseña.
- Debe de comparar si un usuario existe o no dentro de la base de datos, y dependiendo de esto, el usuario podrá ingresar a la siguiente ventana.
- Podrá ser capaz de elegir las opciones de comida del restaurante elegido, ya sea la comida, bebida, postres, etc.
- Le mostrará en pantalla la cantidad de dinero que le cobraron en quetzales.
- El usuario podrá ser capaz de ingresar la zona en la que se encuentra y el programa deberá de mostrar en pantalla la franquicia más cercana para que pase a recoger su pedido.
- Y por último, le mostrará opciones de un link para escuchar una lista de recomendación musical y así poder disfrutar su trayecto mientras está en el tráfico.

III. IDENTIFICACIÓN Y DESCRIPCIÓN DE CLASES NECESARIAS

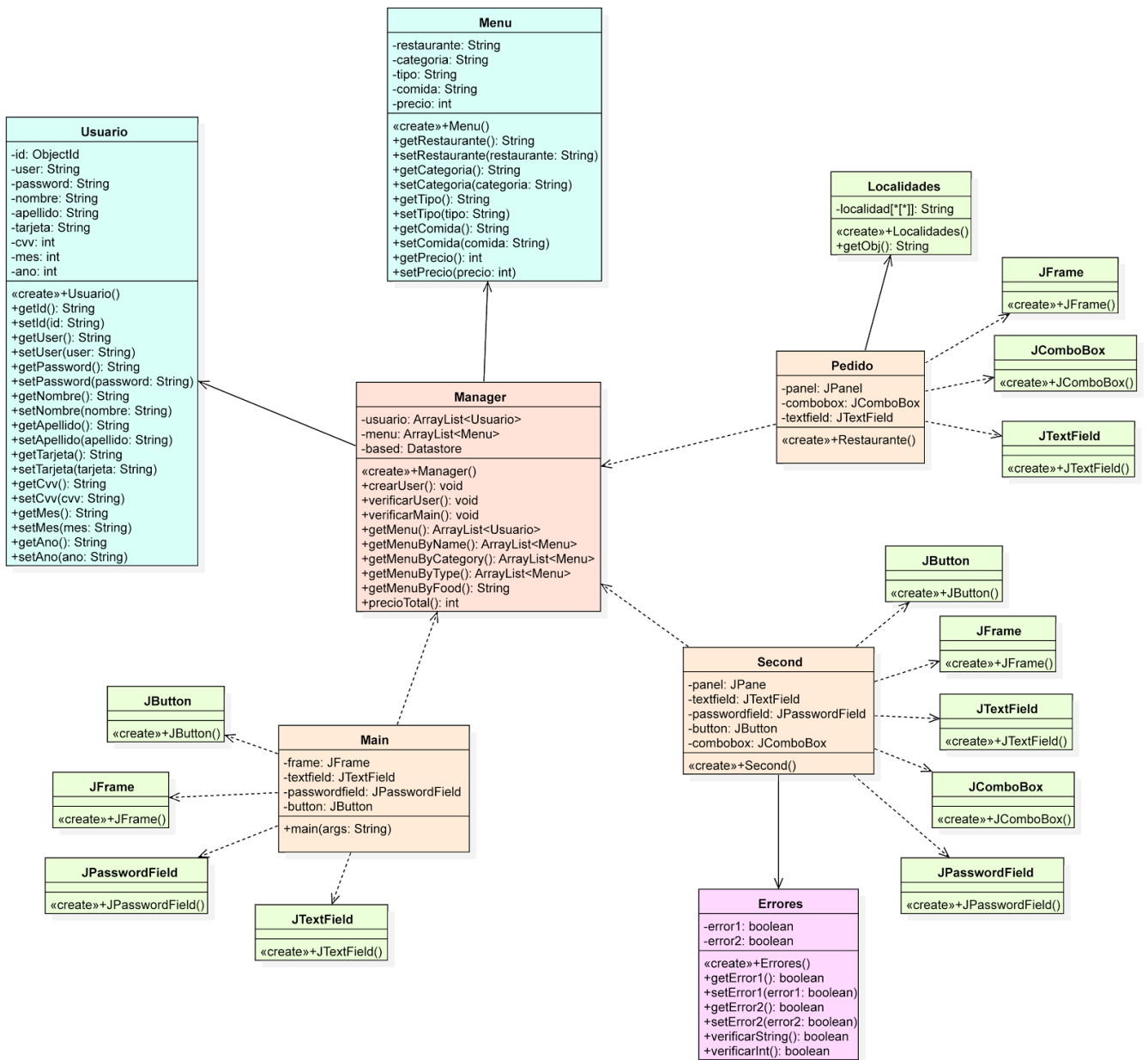
- **Main:** Será capaz de interactuar con el usuario; sin embargo, no debe de realizar ninguna operación. Aquí se deberá de importar la clase Scanner para que se puedan leer los datos de las dos clases antes mencionadas. En esta clase se implementará la GUI principal, la cual contiene el acceso del usuario a la cuenta de iOrder.
- **Manager:** Esta clase tiene como propósito realizar los métodos necesarios para unir la base de datos con la clase Usuario e implementar los requisitos funcionales. En su

constructor se manda a llamar MongoDB y Morphia, por lo cual, se crean objetos de este tipo. En esta clase se crea la base de datos.

- Atributos:
 - Usuario: Es un ArrayList de tipo Usuario y su función es ir guardando todos las cuentas creadas en la base de datos a una lista.
 - Based: Es de tipo Datastore y su función es empezar a crear una conexión con la base de datos.
- Métodos:
 - crearUser: Es un método void usado para crear usuario y guardarlo en la base de datos
 - verificarUser: Es un método tipo boolean usado para la verificar el ingreso de un usuario ya existente.
 - verificarMain: Método tipo boolean usado para verificar si un usuario ya existe cuando está ingresando en la página de inicio (login).
- **Usuario:** Esta clase tiene como principal objetivo guardar los datos del usuario. Se creó el constructor, junto con los getters y setters de cada atributo (no contiene métodos ni toString). En esta clase se crea el @Entity para hacer la conexión con la base de datos.
 - Atributos:
 - id: Es un atributo de tipo ObjectId.
 - User: Es de tipo String y tiene como función guardar el nombre de usuario.
 - Password: Es de tipo String y tiene como función guardar la contraseña del usuario.
 - Nombre: Es de tipo String y tiene como función guardar el nombre de la persona de nuevo ingreso a la cuenta.
 - Apellido: Es de tipo String y tiene como función guardar el apellido.
 - Tarjeta: Es de tipo String y tiene como función guardar el número de tarjeta de crédito/débito.
 - CVV: Es de tipo int y tiene como propósito guardar el número de seguridad de la tarjeta.
 - Mes: Es de tipo int y su función es guardar el mes en el que se expira la tarjeta.
 - Año: Es de tipo int y su función es guardar el año en el que se expira la tarjeta.
- **Menu:** Esta clase tiene como principal objetivo crear un array de los datos de cada restaurante. Esto es: Restaurante, categoría, tipo, comida y precio. Se creó el constructor, junto con los getters y setters de cada atributo.
 - Atributos:
 - Restaurante: tipo String, guarda el nombre del restaurante.

- categoría: tipo String, guarda el tiempo de la comida (desayuno, almuerzo o cena).
 - tipo: tipo String, guarda que tan fuerte es la comida (Light, mid o completa).
- **Errores:** Esta clase tiene como objetivo crear los métodos necesarios para los posibles errores dentro del programa.
 - Atributos:
 - error1: Es un atributo tipo boolean.
 - error2: es un atributo tipo boolean.
 - Métodos:
 - verificarString: Método para verificar si el dato ingresado es un String. Este método es tipo boolean.
 - verificarInt: Método para verificar si el dato ingreado es un int. Este método es tipo boolean.
- **Second:** Es una interfaz gráfica que usa los métodos de Manager para crear una cuenta de usuario nueva. En la clase se usa el JFrame, JPanel, JTextField, JComboBox, JLabel y un JButton.
- **Pedido:** Es una interfaz gráfica que permite al usuario escoger la comida y la cantidad de ella para ordenar. Así mismo le muestra al usuario el restaurante más cercano para recoger su orden. En la clase se usa el JFrame, JPanel, JTextField, JComboBox, JLabel y un JButton. Le da opciones de música al usuario.

IV. DISEÑO DEL SISTEMA



V. INTERACCIÓN CON EL USUARIO

Fecha en la que se le mostró	Datos del usuario	Descripción de requisitos funcionales mostrados	Comentarios y sugerencias hechas por el usuario	Elementos a incorporar
22/10/2017	Maryandré Salguero 17032	Se mostró la consola del programa (aún no había interfaz gráfica)	Se ve desordenado al momento de mostrar todo el menú.	Se implementará GUI para mostrar todos los datos del menú más ordenados.
22/10/2017	Sebastian Tobar 17147	Se mostró la consola del programa (aún no había interfaz gráfica)	No entiendo lo que me están mostrando. Es muy difícil identificar el precio en cada comida.	El precio no saldrá hasta que haya completado la orden.
22/10/2017	Ana Lucia Pinto 17138	Se mostró la consola del programa (aún no había interfaz gráfica)	Que buena idea, pero no es muy amigable con el usuario. Mejoren eso.	Implementar GUI para que sea más agradable con el usuario.
22/10/2017	Marco Villatoro 17770	Se mostró la consola del programa (aún no había interfaz gráfica)	Está bien la idea, solo que deberían de colocar progra defensiva al momento de crear una cuenta.	Al momento de agregar GUI, implementar JOptionPane para mostrar el error y la manera de solucionarlo.
05/11/2017	Chris Ramirez 15565	Creación de cuenta (nombre, tarjeta de crédito, usuario, contraseña, etc).	Agregar botón regresar en la pantalla para ingresar datos.	Agregar botón de regresar en las pantallas.
	Diego Soler 15415	Creación de cuenta (nombre, tarjeta de crédito,	Me gusta mucho el diseño, solo que sería bueno que	Organizar pantalla de obtención de datos, hacer

05/11/2017		usuario, contraseña, etc).	agregaran más explícito el campo para llenar los datos.	mención específica de datos a solicitar.
05/11/2017	Antonio Ixtococ 15582	Creación de cuenta (nombre, tarjeta de crédito, usuario, contraseña, etc).	Organicen la interfaz gráfica. Me cuesta mucho entender lo que me están pidiendo.	Organización de GUI y modificarla de tal manera que sea agradable a la vista.
05/11/2017	Dan Alvarez 14628	Creación de cuenta (nombre, tarjeta de crédito, usuario, contraseña, etc).	Simplifiquen más la GUI. Considero que hay más opciones en Swing, como en la contraseña, deberían de usar un JPasswordField.	Organización de GUI e implementación de JPasswordField para privacidad del usuario.
15/11/2017	Alejandra Camposeco 17126	Mejor orden en GUI. Selección de comida y creación de orden. Link para playlist.	Me gusta mucho lo amigable de la interfaz con el usuario.	Ningún elemento a incorporar.
15/11/2017	Sergio Penagos 15459	Mejor orden en GUI. Selección de comida y creación de orden. Link para playlist.	Me gustaría que tuvieran más opciones de restaurante. La manera de pedir está muy cargada.	Terminar con las demás opciones de restaurantes y separar por áreas los tipos de comida de cada restaurante
15/11/2017	Allyson Camposeco 16117	Mejor orden en GUI. Selección de comida y creación de orden. Link para playlist.	Al momento de pedir comida, se ve muy cargado por mucha información. Sería bueno que lo simplificaran más.	Separar por áreas los tipos de comida de cada restaurante.
15/11/2017	Juan Andrés García 15046	Mejor orden en GUI. Selección de comida y creación de orden. Link para playlist.	Si estuviera manejando y mirara todas las opciones de comida, habría una posibilidad de que chocara.	Separar por áreas los tipos de comida de cada restaurante.

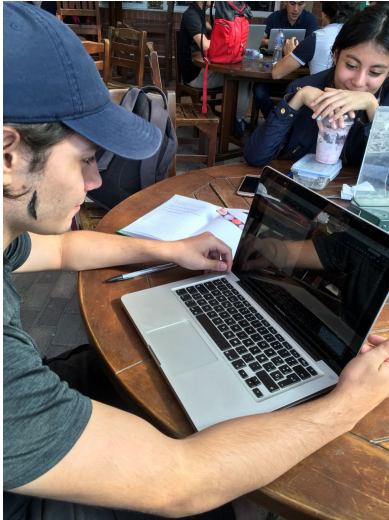


Imagen 1. Primera demostración del programa (22/10/17)



Imagen 2. Primera demostración del programa (22/10/17)

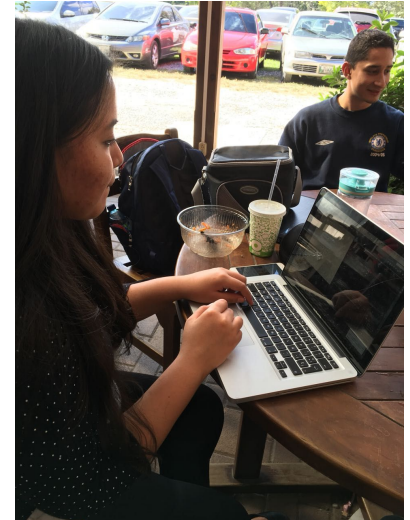


Imagen 3. Primera demostración del programa (22/10/17)



Imagen 4. Segunda mostración del programa (5/11/17)



Imagen 5. Segunda mostración del programa (5/11/17)



Imagen 6. Segunda mostración del programa (5/11/17)

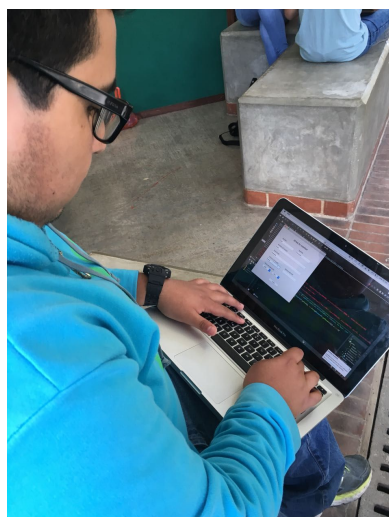
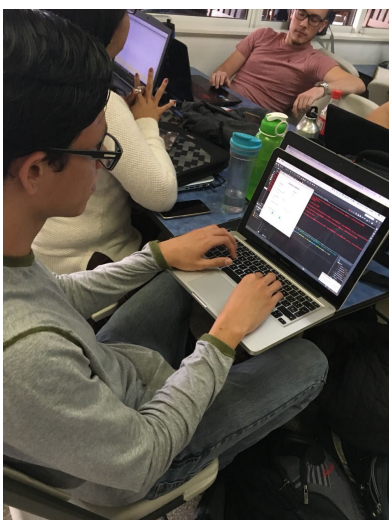


Imagen 7. Tercera mostración del programa (15/11/17)

Imagen 8. Tercera mostración del programa (15/11/17)

VI. IMPLEMENTACIÓN

Observar en los documentos de .java enviados. Ahí se encuentran todas las clases creadas para el proyecto.

VII. PLANIFICACIÓN Y GESTIÓN

Mayra Silva

17276

Fecha	Inicio	Fin	Tiempo interrupción	Delta Tiempo	Tarea	Comentarios
31/10/2017	9:30 pm	11:00 pm	20 minutos	1 hora y 30 minutos	Crear conexión con la base de datos	Subí a Blackboard tarea de cálculo
9/11/2017	11:20 pm	11:40 pm	0 minutos	20 minutos	Subir la conexión de la base de datos a GitHub	El trabajo estaba muy sencillo
10/11/2017	4:00 pm	10:00 pm	4 horas	6 horas	Mejorar el diseño de la GUI	Estaba en el trabajo; estaba ocupada
12/11/2017	9:50 am	1:00 pm	1 hora	3 horas y 10 minutos	Creación de excepciones y modificar la matriz	Fui a una tutoría de Física
14/11/2017	10:30 am	11:15 am	10 minutos	45 minutos	Aplicación de progra defensiva	Fui a Gitane a comprar comida
15/11/2017	12:40 pm	1:20 pm	10 minutos	40 minutos	Creación de la GUI para ordenar comida	Comprar mi almuerzo

Odalís Reyes

17032

Fecha	Inicio	Fin	Tiempo	Delta	Tarea	Comentarios
-------	--------	-----	--------	-------	-------	-------------

			interrupción	Tiempo		
31/10/2017	5:00 p.m.	8:00 p.m.	1 hora	3 horas	Creación inicio GUI	Fui a comer
4/11/2017	4:30 p.m.	5:00 p.m.	½ hora	1 hora con 30 minutos	Creación de solicitud de datos para el usuario (Second)	Tuve que estudiar para física
12/11/2017	3:00 p.m.	4:00 p.m.	0 minutos	1 horas	Creación clase Dominos	Estaba fácil el diseño de la GUI
13/11/2017	2:07 p.m.	4:13 p.m.	2 horas con 6 min	15 min	Implementación de base de datos a usuarios	Tuve que almorzar
14/11/2017	8:00 p.m.	10:00 p.m.	2 horas	25 min	Modificación Main, Manager y Second	Tuve que tomar una power nap
15/11/2017	12:00 p.m.	3:00 p.m.	30 minutos	3 horas	Verificación de usuario en el Login y en Second	Fui a comprar mi almuerzo a Mac

Ivan Maldonado

17211

Fecha	Inicio	Fin	Tiempo interrupción	Delta Tiempo	Tarea	Comentarios
31/10/2017	3:00 p.m.	4:30 p.m.	30 min	1 hora	Modificación matriz de comida	Tuve que ir al baño
7/11/2017	2:54 p.m.	3:24 p.m.	10 min	30 min	Creación de la clase con los links	Tuve que ver video para crear links en java
9/11/2017	5:20 pm	8:00 pm	50 minutos	2 horas con 40 minutos	Creación de la matriz Localidades	Tuve que repasar Química 2
13/11/2017	12:30 pm	1:30 pm	0 minutos	1 hora	Modificación a la matriz de Localidades	Estaba sencilla la modificación
14/11/2017	10:20 am	11:50 am	15 minutos	1 hora con 30 minutos	Modificación a la clase	Fui a comprar mi

					Tercero y agregar la selección de un restaurante	comida a Red
15/11/2017	9:00 a.m.	12:30	2 horas	3 horas y 30 minutos	Creación Clase Burger	Tuve clase de Estadística