

Universidad del Valle de Guatemala  
Programación Orientada a Objetos

**Integrantes:**

Cristian Perez 16011  
Diego Fernandez 16344  
Gadhi Rodriguez 16206  
Rodrigo Urrutia 16139  
Maria Retolaza 16339  
Helmuth Nistal 16104

Fecha de Entrega: 30 de Septiembre de 2016

Sección:10

Catedrática: Lynette

Auxiliares: Santiago Solórzano

## Primera entrega, Fase 1 del proyecto

### Retrospectiva

- **Problema:**

Actualmente Guatemala sufre de contaminación, tanto de individuos, como de empresas. En este trabajo se cree que es importante cuidar los recursos naturales, por lo tanto se desarrolla este proyecto para promover que las personas y empresas reciclen. Una de las razones por las cuales las personas no le ponen importancia al reciclaje es porque no conocen lugares que acepten materiales de segunda mano para reciclar. Como no conocen los principios de reciclaje tampoco entienden las ventajas que este proceso tiene.

- **Necesidades e Insights (Oportunidades)**

Al realizar las entrevistas a diferentes tipos de posibles usuarios, teniendo claro que nuestro objetivo era promover el cuidado al medio ambiente y la practicar el reciclaje, observamos que uno de los mayores problemas es que la mayoría de los entrevistados no mostraba interés a la importancia del reciclaje. Muchos de ellos demostraron no tener el cuidado correcto para desechar su basura, teniendo que quemarla o tirarla en lugares indebidos. Es por esta razón que llegamos a la conclusión de que debíamos encontrar una forma de promover el debido desecho de materiales usados. Nuestra primera idea era crear una máquina que regalar algo de uso diario a cambio de papel o latas recicladas. Al formular nuevas entrevistas supimos que no a todas las personas les gusta lo mismo, así que llegamos al acuerdo de crear una aplicación que beneficiaría al usuario de forma económica enseñándole cuáles eran los lugares más cercanos a su posición actual para llevar su material de reciclaje.

- **Descripción de la Fase de Ideación e ideas más votadas**

Realizando una lluvia de ideas para crear nuestra aplicación, supimos que queríamos ayudar al medio ambiente de alguna manera. Primero buscamos algo más físico, una máquina que regalara cualquier producto de consumo diario de un estudiante a cambio de papel o latas recicladas. Luego decidimos contribuir al medio ambiente por medio de una aplicación que ayudará al usuario a saber cuál era la recicladora más cercana y que fuera formando el hábito de reciclar a través de información importante en relación al tema.

- **Descripción de Prototipos**

Se realizaron varios dibujos de nuestro proyecto, llegando a formar parte del Storyboard antes de realizar la aplicación. Esta última la mostramos a distintas personas para que nos dieran su opinión acerca de los aspectos positivos, negativos e interesantes de nuestro proyecto. De esta manera nos mencionaron cómo mejorar esos aspectos negativos para que la interfaz fuera lo más amigable posible. Luego se realizó un segundo prototipo, desarrollado en Python y sin interfaz gráfica, para idear los algoritmos y la programación necesaria para poner nuestras ideas nuestro programa a prueba. Por último se realizó el prototipo final utilizando Kivy para realizar la interfaz gráfica y Python para la programación principal.

**Planificación.**

- Desglose del proyecto en tareas con su respectivo responsable y fecha propuesta de terminación

**Problema**-----**(Todos)**

**Planificacion**-----**(Todos)**

**Requisitos funcionales**-----**(Todos)**

**revisar el trabajo**-----**(Cristian y María)**

**obtener lugares de reciclar**-----**(Ghadi)**

**precios por reciclaje**-----**(Diego)**

**investigar datos curiosos**-----**(Rodrigo)**

**diseñar interfaz gráfica** -----**(Helmuth)**

**realizar algoritmo narrativo** -----**(Gadhi)**

**Mejorar el algoritmo de búsqueda de recicladoras**-----**(Gadhi y Helmuth)**

**requisitos funcionales**-----**(Todos)**

**Análisis y descripción de clases, métodos y atributos**-----**(Todos)**

**desarrollar uml**-----**(Rodrigo y María)**

**Prototipo 1(Diseñar)**-----**(Helmuth)**

**Revisar Prototipo 1**-----**(Todos)**

**Prototipo 2(Implementar información)**-----**(Todos)**

**Revisar Prototipo 2**-----**(Todos)**

**Investigar cómo conectar java vrs sql**-----**(Diego y María)**

**diseño tablas -----(Cristian)**

**Prototipo 3-----**

**Revisar prototipo 3----- (Todos)**

**codigo conectar base datos-----**

## Formulario

**Nombre:** Pablo Gadhi Rodriguez

Semestre II – 2016

**Carné:** 16206

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
03/09/2016	8:00		N/A	120	Requisitos funcionales	
03/09/2016	8:00		N/A	120	Análisis y descripción de clases, métodos y atributos	
03/09/2016	8:00		N/A	120	Necesidades, descripción de prototipos, fase de ideación e ideas más votadas.	
23/08/16	5:00 p.m	7:00 p.m.	30min	2 horas	Actualizar información de recicladoras de guatemala.	
24/08/16	7:50 a.m.	10:15 a.m	30 min	2 horas	Algoritmo narrativo.	
20/09/16	5:00 p.m	7:00 p.m	45 min	2 horas	Mejorar algoritmo de	

					búsqueda de recicladoras.	
21/09/16	5:00 p.m	7:00 p.m	1 hora	2 horas	Revisar Prototipo 1	
30/09/2016	7:50 am	10:00 am	2 horas	2 horas	Corregir errores y comentar	

**Nombre:** Cristian Pérez

Semestre II – 2016

**Carné:** 16011

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
19/08/2016	4:00	4:30	10 mins	1/2 hora	Requisitos funcionales	
02/09/2016	3:00	3:30	18 mins	1/2 hora	Análisis y descripción de clases, métodos y atributos	
09/09/2016	2:00	2:30	15 mins	2 horas	revisar el trabajo	
15/09/2016	5:00	7:00	10 mins	2 horas	diseño tablas	
17/09/2016	11:00	3:00	10 mins	4 horas	Hacer menú	
25/09/2016	10:00	12:00	25 mins	2 horas	Revisión trabajo	
27/09/2016	4:00	7:00	0 mins	3 horas	Unión paneles	

**Nombre:** Rodrigo Urrutia

Semestre II – 2016

**Carné:** 16139

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Definir problema	Discutir el problema a solucionar
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar planificación	Ordenar las tareas para cada integrante
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar requisitos funcionales	Discutir las características del proyecto
20/08/2016	21:00	22:00	N/A	60	Investigar datos curiosos	
03/09/2016	14:00	16:00	N/A	120	Requisitos funcionales	
03/09/2016	16:00	18:00	N/A	120	Análisis y descripción de clases, métodos y atributos	
09/09/2016	2:00	2:30	15 mins	09/09/2016	Desarrollar UML	
15/09/2016	5:00	7:00	10 mins	15/09/2016	Desarrollo de GUI	
17/09/2016	11:00	3:00	10 mins	17/09/2016	Desarrollo de GUI	
25/09/2016	10:00	12:00	25 mins	25/09/2016	Desarrollo de GUI	

**Nombre: Diego Fernandez**

Semestre II – 2016

**Carné: 16344**

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Definir problema	Discutir el problema a solucionar

19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar planificación	Ordenar las tareas para cada integrante
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar requisitos funcionales	Discutir las características del proyecto
19/08/2016	11:00	11:30	N/A	30	<i>precios por reciclaje</i>	Investigar sobre los precios pagados por las mayores recicladoras.
25/08/2016	7:45	10:15	20 min	120 min	Revisar prototipo 1	Todos los integrantes revisamos el prototipo 1.
27/09/2016	15:00	16:20	15 min	60 min	<i>Investigar cómo conectar java vrs sql</i>	Entender realmente como conectar java a sql
30/09/2016	7: 45	10:15	20 min	75 min	Modificar algoritmo para las zonas	Modificar la proximidad de las zonas conforme las mayores avenidas y el mapa actual de Guatemala.
30/09/2016	7: 45	10:15	15 min	15 min	Programación defensiva	Revisar la programación defensiva

**Nombre:** Helmuth Nistal

Semestre II – 2016

**Carné:** 16104

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Definir problema	Discutir el problema a solucionar
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar planificación	Ordenar las tareas para cada integrante
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar requisitos funcionales	Discutir las características del proyecto
20/08/16	9:00 pm	9:35 pm	5min	30min	diseñar interfaz gráfica	Realizar un ejemplificación del programa
25/08/16	9:00 pm	9:35 pm	5min	30min	Mejorar algoritmo	Mejorar el algoritmo de búsqueda de recicladoras

31/08/16	8:00 pm	9:10 pm	10min	60min	análisis y descripción	de clases métodos y atributos
31/08/16	9:00 pm	9:35 pm	5min	30min	Plantear prototipo 1	Crear una estrategia para desarrollar el prototipo
7/09/16	3:00 pm	3:30 pm	0min	30min	revisar prototipo 1	Discutir resultado del prototipo
16/09/16	9:00 pm	9:35 pm	5min	30min	prototipo 1	Iniciar a realizar el prototipo
21/09/16	3:00 pm	3:30 pm	0min	30min	Organizar las ideas de cada integrante	Organizar las ideas para facilitar la unión de todo los trabajos
28/09/16	3:00 pm	3:30 pm	0min	30min	revisar con el grupo el prototipo 1	Discutir con el grupo resultado del prototipo
30/09/16					Fecha entrega	

**Nombre:** Maria Mercedes Retolaza Reyna

Semestre II – 2016

**Carné:** 16339

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Definir problema	Discutir el problema a solucionar
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar planificación	Ordenar las tareas para cada integrante
19/08/16	7:50 a.m.	10:15 a.m.	25 min	2 horas	Realizar requisitos funcionales	Discutir las características del proyecto
22/08/16	7:00 p.m.	8:00 p.m.	15 min	1 hora	Análisis y descripción de	Se realizó una descripción de las primeras clases y métodos que se

					clases, métodos y atributos	tenían a la fecha. Esto con el fin de mantener un control.
23/08/16	5:00 p.m	7:00 p.m.	30min	2 horas	Diseño UML	Se realizó una análisis nuevo en el UML, esto se realizó debido a que no estábamos seguros si implementar todas las clases o no.
24/08/16	7:50 a.m.	10:15 a.m	30 min	2 horas	Investigar cómo conectar java vrs sql	Para la última entrega, queremos convertir nuestro programa en una base de datos para almacenar los datos del usuario y recicladoras.
20/09/16	5:00 p.m	7:00 p.m	45 min	2 horas	Revisar Prototipo	Revisé lo que correspondía del prototipo para implementar alguna nueva clase.
21/09/16	5:00 p.m	7:00 p.m	45 min	2 horas	Realización de clase conversiones y controlador de conversiones.	Se realizó la clase conversiones para calcular las equivalencias de la cantidad que recicle a datos curiosos (Conversiones).
22/09/16	5:00 p.m	8:00 p.m	1 hora	3 horas	Revisión de análisis	Se realizó la revisión del análisis

**Formulario 1.** Gestión del tiempo en el cumplimiento de las tareas planificadas

**Tabla 2.** Objetivos e instructivo para llenar el formulario 1

### **Lista de Requisitos Funcionales**

- Registrarse (crear cuenta)
- Dos tipos de usuario (Empresa , Usuario individual)
- Proteger la identidad del usuario
- Guardar la información del usuario en una base de datos
- Guardar la información de las recicladoras en una base de datos
- Menú principal de bienvenida
- Diseño interactivo con logotipo propio
- Encontrar recicladoras por categoría
- Encontrar la recicladora más cercana
- Crear una base de datos en donde se guardará toda la información recolectada del usuario
- Calcular cantidad reciclada en función de seres vivos salvados
- Calcular dinero ahorrado por rehusar en lugar de consumir recursos
- Calcular dinero recibido por recicladoras
- Datos curiosos en el menú de bienvenida



- Programación defensiva (validaciones)

Identificación de requisitos funcionales:

Partiendo de las necesidades y oportunidades identificadas, elabore una lista de los requisitos funcionales que tendrá su sistema (Lo que debe hacer el sistema). Tenga en cuenta los requerimientos de seguridad que tendría que tener su aplicación, tipos de usuario por ejemplo (administrador, no administrador, etc.)

Identificación y Descripción de Clases necesarias:

Detecte las clases que necesita para modelar la solución del problema. Recuerde que debe usar el patrón Modelo- Vista-Controlador, lo que significa que las clases con las que tenga interacción el usuario no deben contribuir a la lógica de lo que hace la aplicación, únicamente se utilizan para obtener información del usuario y mostrarle resultados. Así mismo las clases Modelo y Controlador, no pueden mostrar directamente resultados al usuario.

Para cada una de las clases elabore una descripción donde especifique para que se usará, así como el propósito de atributos y métodos.

**Nota:** Como dividimos nuestro proyecto en dos entregas, únicamente cumplimos con la mitad de los requisitos funcionales. Esto se debe a que la siguiente parte se necesita con la base de datos que estaremos implementando para la entrega siguiente.

### Descripción de clases del programa:

#### 1. Recicladora

- a. Propósito
  - i. Posee todos los atributos necesarios para crear un objeto de una recicladora
- b. Atributos
  - i. Tiene id, nombre, direccion, telefono, pagina, zona, papel, plastico, metal y vidrio. Así se puede ubicar la localización de la recicladora y se especifica que tipo de recicladora es. (plástico, metal o vidrio)
- c. Métodos
  - i. Contiene sets y gets para cada atributo para poder guardar los objetos de recicladoras y mostrarselos al usuario. También tiene un toString que muestra todos los atributos en el orden establecido arriba en atributos.

#### 2. CiudadVerde

- a. Propósito
  - i. Sirve como controlador para la búsqueda de recicladoras. Crea 17 objetos de tipo "Recicladora" con los atributos específicos de cada recicladora y los ingresa en un arreglo para facilitar su búsqueda.
- b. Atributos

- i. 17 Objetos de tipo “Recicladora”, especificando los atributos de cada recicladora en el constructor.
  - ii. Arreglo de Recicladoras.
  - iii. Arreglo bidimensional de enteros, que funcionará para que el programa decida la recicladora más cercana que mostrará en caso de que no exista en ninguna zona.
- c. Métodos
  - i. Método que busca la recicladora más cercana basándose en la zona y el material ingresada en los atributos de dicho método. Devuelve la información específica de la recicladora que encontró, basándose en un tercer atributo que especifica lo que se quiere mostrar.

### 3. Gui

- a. Propósito
  - i. Sirve para interactuar con el usuario y mostrarle resultados de manera gráfica y también es amigable, utilizando combo boxes y un JPasswordField para proteger la contraseña del usuario.
- b. Atributos
  - i. A través de JButtons y JComboBox se implementan los métodos de la clase conversiones. Hay más de un panel para permitir que el usuario vea un panel a la vez.
- c. Métodos
  - i. Los botones poseen action listeners que cambian la visibilidad de los paneles, el menú es el panel que está visible al principio y si se presiona el botón de registro, se cambia la visibilidad del primero y el panel de registro se vuelve visible.

### 4. Conversiones

- a. Propósito
  - i. El propósito de esta clase es contener todas las operaciones que se utilizan en el programa.
- b. Atributos
  - i. La clase no tiene ningún atributo, debido a que está directamente conectada con la GUI
- c. Métodos
  - Los métodos para la clase representa a cada una de las conversiones que vamos a realizar. A continuación se describe cada método de la clase:
    - i. `public static double kgAlbs(double cant):` Conversión de Kilogramos a Libras.
    - ii. `public static double kilosAarboles(double kilo):` Conversión de Kilogramos a Árboles.
    - iii. `public static double arbolesAlibros(double arb):` Conversión de Árboles a Libros.
    - iv. `public static double kilosAtoneladas(double kg):` Conversión de Kilos a toneladas
    - v. `public static double toneladasAlouvre(double toneladas):` Conversión de toneladas a Alouvre, esto se utiliza para la conversión de vidrio.

- vi.     public static double metalAEiffel(double toneladas): Conversión de metal a Torre Eiffel.
- vii.    public static double barby(double kg): Conversión de barbie a kilogramo.

#### 4. ControladorConversiones

##### d. Propósito

- i.     El propósito de esta clase es realizar los métodos para unir los datos obtenidos en conversiones con los datos que se recolectarán en el programa.

##### e. Atributos

- i.     La clase no tiene ningún atributo, debido a que está directamente conectada con la GUI

##### f. Métodos

- Los métodos para la clase representa a cada una de las conversiones que vamos a realizar. A continuación se describe cada método de la clase:

Esta clase contiene dos librerías:

- **import java.util.HashMap;** // es una colección de objetos, (como los Arrays), pero estos no tienen orden. Cada objeto se identifica mediante algún identificador apropiado.
- **import java.util.Map;** // La Interface Map (java.io.Map) en Java, nos permite representar una estructura de datos para almacenar pares "clave/valor"; de tal manera que para una clave solamente tenemos un valor.
- i.     public static Map<String, Double> calcPaper(double cant, String lb\_kg): Recibe la cantidad de arboles y de libros, dando el resultado para la GUI.
- ii.    public static double calcMetal(double cant, String lb\_kg): Recibe la cantidad y devuelve los resultados.
- iii.   public static double calcPlastic(double cant, String lb\_kg): Recibe la cantidad y devuelve los resultados
- iv.    public static double calcGlass(double cant, String lb\_kg): Recibe la cantidad y devuelve los resultados