



Propuesta para situación problema

Equipo 6

Reynaldo Hernández González

A00829814

Diego Novoa Padilla

A01620597

Iván López

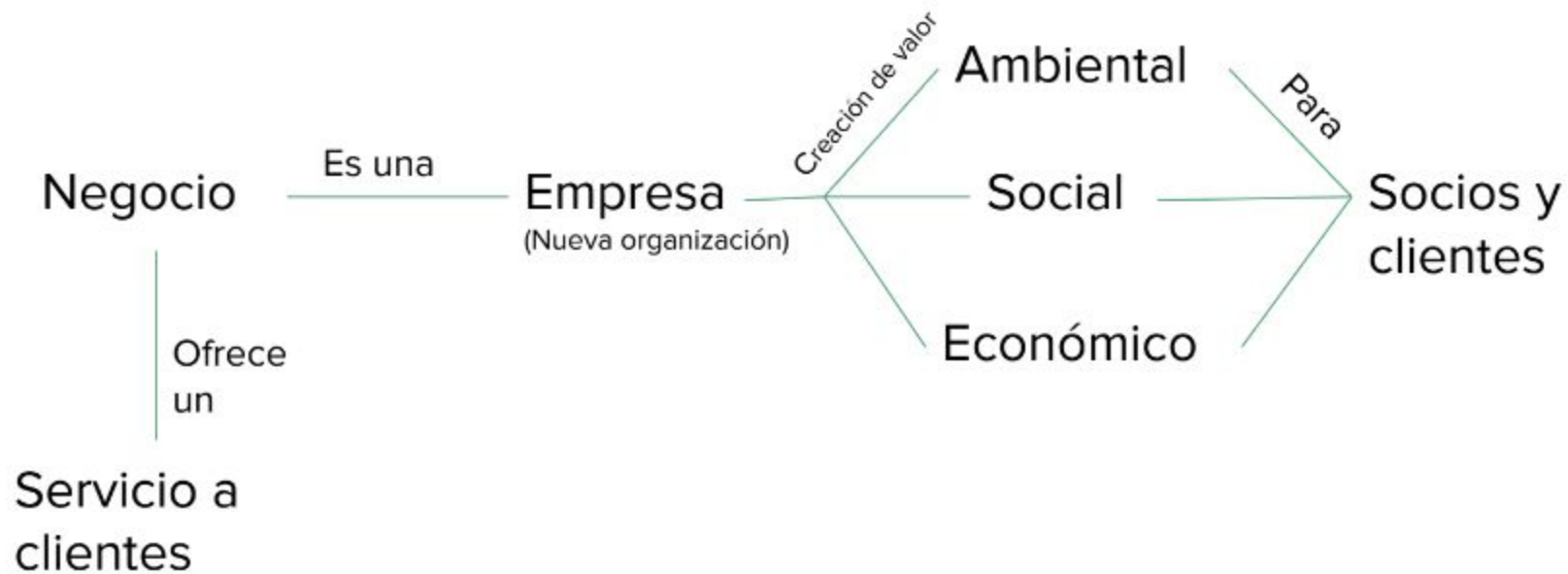
A01284875



Reacción de comercios locales ante la pandemia

- Ofrecer servicios de manera virtual
- Adaptar productos a la situación de la pandemia (ej. hacer gel antibacterial)
- Usar entregas sin contacto para que sus servicios estén disponibles.
- Pedir a los empleados que aprendan nuevas habilidades para respaldar los cambios en los modelos comerciales.
- Adoptar nuevas fuentes de ingresos, como convertir las líneas de producción para fabricar desinfectante para manos, que tiene una gran demanda.
- Instituir nuevas medidas de seguridad, como instalar barreras de plexiglás entre el personal y los clientes.
- Adopción de nuevos procesos tecnológicos

Negocio de enfoque: venta de productos





Características de dueños y clientes

Dueños

- compran o hacen un producto para vender
- tienen que surtirse
- algunos pueden hacer entregas a domicilio

Cliente

- quieren la seguridad de que pueden obtener lo que buscan en el momento
- precios accesibles
- generalmente buscan algo de calidad
- aprecian la facilidad de conseguir un producto (accesibilidad del local, servicio a domicilio seguro)

Propuesta: aplicación de navegación que genera una *ruta optimizada* para la entrega de productos

Clientes
Productos a buen precio y a domicilio

Pan a buen precio y a domicilio

Consciencia de consumo y producción responsable

Necesidad:

Acercamiento:

App

Socios

Aumentar ventas

Mayor visibilidad



Requerimientos principales

- Domicilio del cliente
- Dispositivo con GPS



Diseño de funcionalidad

1. Varios usuarios realizan compras en línea para obtener sus productos por domicilio
2. La aplicación toma los datos de las ubicaciones de cada cliente. Después de cierto horario, deja de tomar datos y se pone a calcular la ruta más óptima
3. El empleado sigue la ruta optimizada y hace las entregas
4. Los clientes tienen acceso a la ubicación del empleado en todo momento para tener una idea de cuándo llegará a su ubicación



Análisis para el desarrollo

1. Crear una base de datos para guardar información del cliente y los pedidos
2. Desarrollar un programa que procesa los datos de ubicación para generar una ruta óptima
3. Hacer una aplicación con versiones para el cliente y para el dueño
 - a. Cliente crea una cuenta e ingresa datos personales para hacer compras en línea, y puede monitorear posición de vehículo de entrega cada 5 minutos
 - b. Dueño recibe la información de los clientes y sus pedidos, la ruta optimizada, y puede monitorear posición de vehículo de entrega en tiempo real



Prototipo

Datos recibidos:

Cliente: Arturo

Domicilio: Avenida Eugenio Garza Sada 2640

Pedido: F16

Cliente: Carlos

Domicilio: Avenida Luis Elizondo 250

Pedido: A14

Cliente: Laura

Domicilio: Avenida Junco de la Vega 2602

Pedido: D19

Orden de paradas:

1) Cliente: Carlos

Domicilio: Avenida Luis Elizondo 250

Pedido: A14

2) Cliente: Laura

Domicilio: Avenida Junco de la Vega 2602

Pedido: D19

3) Cliente: Arturo

Domicilio: Avenida Eugenio Garza Sada 2640

Pedido: F16



Código

```
#include <iostream>
#include <vector>

using namespace std;

class Client {
private:
    string name;
    int location;
    string order;
    string address;

public:
    Client() {
        name = "null";
        location = 0;
        order = "null";
        address = "null";
    }

    Client(string _name, int _location, string _order, string _address) {
        name = _name;
        location = _location;
        order = _order;
        address = _address;
    }
}
```

```
void setName(string _name) {  
    name = _name;  
}  
  
void setLocation(int _location) {  
    location = _location;  
}  
  
void setOrder(string _order) {  
    order = _order;  
}  
  
void setAddress(string _address) {  
    address = _address;  
}  
  
string getName() {  
    return name;  
}  
  
int getLocation() {  
    return location;  
}  
  
string getOrder() {  
    return order;  
}  
  
string getAddress() {  
    return address;  
}  
};
```

```

50  /*
51  Mezcla/combina dos subarrays de un arr.
52  Se utiliza para la función optimize().
53
54  Entradas: vector<int>* arr; un array de tipo Client
55           |      |      int const left; índice del lado izquierdo de un array principal
56           |      |      int const mid; índice del punto medio de un array principal
57           |      |      int const right; índice del lado derecho de un array principal
58  Salida: N/A
59
60  Complejidad de tiempo: O(n)
61  */
62  void merge(vector<Client>* arr, int const left, int const mid, int const right) {
63      auto const subArr1 = mid - left + 1;
64      auto const subArr2 = right - mid;
65
66      // Crear arrays temporales
67      vector<Client> l_arr, r_arr;
68
69      // Copiar datos arr arrays temporales l_arr[] y r_arr[]
70      for (auto i = 0; i < subArr1; i++)
71          l_arr.push_back(arr->at(left + i));
72      for (auto j = 0; j < subArr2; j++)
73          r_arr.push_back(arr->at(mid + 1 + j));
74
75      auto indexArr1 = 0,
76          indexArr2 = 0;
77      int indexMainArr = left;
78

```

```
79 // Mezclar arrays temporales arr array principal
80 while (indexArr1 < subArr1 && indexArr2 < subArr2) {
81     if (l_arr[indexArr1].getLocation() <= r_arr[indexArr2].getLocation()) {
82         arr->at(indexMainArr) = l_arr[indexArr1];
83         indexArr1++;
84     }
85     else {
86         arr->at(indexMainArr) = r_arr[indexArr2];
87         indexArr2++;
88     }
89     indexMainArr++;
90 }
91
92 // Copiar elementos restantes si es que quedan
93 while (indexArr1 < subArr1) {
94     arr->at(indexMainArr) = l_arr[indexArr1];
95     indexArr1++;
96     indexMainArr++;
97 }
98
99 // Copiar elementos restantes si es que quedan
100 while (indexArr2 < subArr2) {
101     arr->at(indexMainArr) = r_arr[indexArr2];
102     indexArr2++;
103     indexMainArr++;
104 }
105 l_arr.clear();
106 r_arr.clear();
107 }
108
```

```

120  /*
121  Ordena un arreglo por el metodo de unión (merge)
122  donde se compara un par de numeros para
123  cambiarlos de posicion si el número de la izquierda
124  es mayor al número de la derecha.
125
126  Entrada: vector<Client>* arr; un arreglo de tipo Client
127  Salida: N/A
128
129  Complejidad de tiempo: O(nlog(n))
130  */
131  void optimize(vector<Client>* arr, int const start, int const end) {
132      if (start >= end)
133          return;
134
135      auto mid = start + (end - start) / 2;
136      optimize(arr, start, mid);
137      optimize(arr, mid + 1, end);
138      merge(arr, start, mid, end);
139  }
140
141  void print_data(vector<Client>* arr) {
142      for (int i = 0; i < arr->size(); i++)
143          cout << "Cliente: " << arr->at(i).getName()
144              << "\nDomicilio: " << arr->at(i).getAddress()
145              << "\nPedido: " << arr->at(i).getOrder() << "\n\n";
146      cout << endl;
147  }
148
149  void print(vector<Client>* arr) {
150      for (int i = 0; i < arr->size(); i++)
151          cout << i+1 << ") Cliente: " << arr->at(i).getName()
152              << "\nDomicilio: " << arr->at(i).getAddress()
153              << "\nPedido: " << arr->at(i).getOrder() << "\n\n";
154      cout << endl;
155  }

```



```
142 int main() {
143
144     Client cliente1 = Client("Carlos", 1, "A14");
145     Client cliente2 = Client("Laura", 2, "D19");
146     Client cliente3 = Client("Arturo", 3, "F16");
147
148     vector<Client> clientes;
149
150     clientes.push_back(cliente3);
151     clientes.push_back(cliente1);
152     clientes.push_back(cliente2);
153
154     cout << "Datos recibidos:" << endl;
155     print_data(&clientes);
156     optimize(&clientes,0,clientes.size()-1);
157     cout << "Orden de paradas: " << endl;
158     print(&clientes);
159
160     return 0;
161 }
```

Referencias Bibliográficas:



-Se usaron elementos de un trabajo anterior (Pan al instante, 2021) elaborado por:

Cesar Vladimir A01570804

Diego Novoa Padilla A01620597

Fernanda Lee A00830974

Guillermo Cepeda A01284015

-Gurchiek, K. (2010, September 19). Small Businesses Get Creative to Survive During the Pandemic. Shrm.

<https://www.shrm.org/hr-today/news/all-things-work/pages/small-businesses-get-creative-to-survive-during-the-pandemic.aspx>

-Fox, M. (2021, May 19). Here's how these small businesses pivoted to survive during the pandemic. Cnbc.

<https://www.cnbc.com/2021/05/19/heres-how-these-small-businesses-pivoted-to-survive-during-the-pandemic.html>

<https://www.cnbc.com/2021/05/19/heres-how-these-small-businesses-pivoted-to-survive-during-the-pandemic.html>

<https://www.shrm.org/hr-today/news/all-things-work/pages/small-businesses-get-creative-to-survive-during-the-pandemic.aspx>