



Tecnológico de Monterrey

“Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales”

Programación de estructuras de datos y algoritmos fundamentales

Profesora: María Valentina Narváez Terán

Bruno Fernando Zabala Peña - A00838627

20/09/2024

Problemática

En un mundo que está en constante movimiento, la industria del food delivery está tomando un impulso bastante importante, pues ofrece ventajas tanto para clientes y proveedores por igual, tales como:

- Abarcar un mercado mucho más amplio sin la necesidad de una expansión física de un negocio.
- Tener acceso a alimentos sin la necesidad de transportarse.
- Ahorro de tiempo al no tener la necesidad de cocinar.
- Mejorar la imagen de un negocio y hacerlo mucho más competitivo.
- Aumentar las ventas.
- Etc.

Sin embargo, y aunque existen herramientas y plataformas que están haciendo cada vez más sencillo adentrarse en este modelo de ventas (Uber Eats, Rappi, etc.), los negocios aún deben contar con una infraestructura o sistemas que le permitan llevar un monitoreo de lo que implica este servicio, como el registro de ventas o clientes.

En esta primera etapa de la solución, se creó una aplicación capaz de manejar un conjunto de órdenes de food delivery, ordenarlas de manera cronológica y permitir al usuario la búsqueda de datos específicos.

Algoritmos utilizados

La primera función implementada en la aplicación fue el poder organizar de manera eficiente las órdenes del archivo “orders.txt” de acuerdo a su fecha y hora, para lo cual era necesario buscar una forma de asignar un valor a cada orden. Para lograr lo anterior, se creó una función llamada “get_value”:

```

// Funcion para asignar un valor a cada orden de acuerdo a su fecha y hora
int get_value(string order){
    string months[12] = {"ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"};
    string aux;

    int month_end_index = order.find(" ");
    string month = order.substr(0, month_end_index);

    aux = order.substr(month_end_index + 1);

    int day_end_index = aux.find(" ");
    string day = aux.substr(0, day_end_index);

    aux = aux.substr(day_end_index + 1);

    int time_end_index = aux.find(" ");
    string time = aux.substr(0, time_end_index);

    int value = 0;

    // Pasa por todos los elementos del arreglo
    for(int i = 0; i < 12; i++){
        if(month == months[i]){
            // Indice del elemento encontrado
            value += (i + 1) * 100000000;
            break;
        }
    }

    value += stoi(day) * 1000000;

    int hour_end_index = aux.find(":");
    string hour = time.substr(0, hour_end_index);

    aux = time.substr(hour_end_index + 1);

    int minute_end_index = aux.find(":");
    string minute = aux.substr(0, minute_end_index);

    aux = aux.substr(minute_end_index + 1);

    int second_end_index = aux.find(" ");
    string second = aux.substr(0, second_end_index);

    value += stoi(hour) * 10000;
    value += stoi(minute) * 100;
    value += stoi(second);

    return value;
}

```

Esta función recibe un string como parámetro y devuelve un valor entero. Primero, se busca en el string recibido la fecha encontrando el índice en el que aparece el primer espacio y este fragmento de texto es almacenado en una variable. Este proceso se repite hasta que se tienen también el día y la hora. Una vez que se tienen estos datos, se inicializa una variable y se le asigna un valor inicial según el mes encontrado (este se busca en un arreglo que contiene las iniciales de los meses y el índice en el que se encontró es multiplicado por 100000000), a esa luego se le suma el valor del día multiplicado por 100000000, las horas multiplicadas por 10000, los minutos multiplicado por 100 y los segundos.

Una vez que se tiene esta función, se puede pasar a la implementación del algoritmo de ordenamiento. Debido a la gran cantidad de datos que se tiene, se optó

por utilizar un método que tuviera una complejidad temporal baja y finalmente se decidió usar el ordenamiento por fusión (merge sort).

```
void mergeSort(vector<string>& A, int l, int r) {  
    int m;  
  
    if (l < r) {  
        // Encuentra el punto medio  
        m = l + (r - l) / 2;  
  
        mergeSort(A, l, m);  
        mergeSort(A, m + 1, r);  
        merge(A, l, m, r);  
    }  
}
```

```

// Ordenamiento por fusion (merge)
// Complejidad promedio:  $O(n \log(n))$ 
void merge(vector<string>& A, int l, int m, int r) {
    int i, j, k, s1, s2;

    // Tamanos para los subarreglos
    s1 = m - l + 1;
    s2 = r - m;

    // Creación de subarreglos (los vectores tienen tamaño dinamico)
    vector<string> L(s1), R(s2);

    // Copia los elementos al subarreglo de la izquierda
    for(i = 0; i < s1; i++){
        L[i] = A[l + i];
    }

    // Copia los elementos al subarreglo de la derecha
    for (j = 0; j < s2; j++){
        R[j] = A[m + 1 + j];
    }

    i = 0;
    j = 0;
    k = l;

    // Combina los subarreglos
    while(i < s1 && j < s2){
        // Compara los elementos (convierte el string con la fecha a un int) de los subarreglos e inserta el menor
        if(get_value(L[i]) <= get_value(R[j])){
            A[k] = L[i];
            i++;
        }
        else{
            A[k] = R[j];
            j++;
        }
        k++;
    }

    // Inserta los elementos restantes
    while(i < s1){
        A[k] = L[i];
        i++;
        k++;
    }

    while(j < s2){
        A[k] = R[j];
        j++;
        k++;
    }
}

```

Con este método (el cual tiene una complejidad de $O(n \log(n))$), el vector de órdenes es dividido en subarreglos más pequeños (esto hasta llegar a arreglos de un elemento, los cuales por definición ya están ordenados) y forma poco a poco el arreglo ordenado comparando los subarreglos “vecinos” para encontrar e insertar los elementos menores. La función creada recibe un vector de strings, los límites izquierdo y derecho del arreglo y el punto medio del mismo. Para las comparaciones se obtiene el valor de cada orden a través de la función “get_value” y los datos ya

ordenados se almacenan en un archivo de texto (nota: la función no tiene un valor de retorno, pero el vector se ve modificado debido a que este se recibe por referencia).

Para permitirle al usuario la búsqueda de órdenes en fechas específicas, se programaron dos funciones:

```
void single_search(string date, int date_value, vector<string> orders, int n){
    // Archivo en el que se almacenan los resultados de búsqueda
    ofstream search_results("outputs/search_results.txt");

    vector<string> results;
    int i, j;

    // Pasa por todos los elementos del vector de ordenes
    for(i = 0; i < n; i++){
        // Busca coincidencias exactas con la fecha dada
        if(get_value(orders[i]) == date_value){
            // Guarda las coincidencias en el vector de resultados
            results.push_back(orders[i]);
        }
    }

    // Imprime los resultados de la búsqueda y los escribe en el archivo de salida
    cout << "Resultados de la búsqueda:" << endl;
    search_results << "Resultados de la búsqueda \"" << date << "\"\" << endl;

    if(results.size() == 0){
        cout << "No hay resultados para esta fecha";
    }

    else{
        for(j = 0; j < results.size(); j++){
            cout << j + 1 << "- " << results[j] << endl;
            search_results << j + 1 << "- " << results[j] << endl;
        }
    }

    search_results.close();
}
```

```

void range_search(string lower_date, string upper_date, int lower_date_value, int upper_date_value, vector<string> orders, int n){
    // Archivo en el que se almacenan los resultados de búsqueda
    ofstream search_results("outputs/search_results.txt");

    vector<string> results;
    int i, j;

    // Pasa por todos los elementos del vector de órdenes
    for(i = 0; i < n; i++){
        // Verifica si la orden está dentro del rango de fechas
        if(get_value(orders[i]) >= lower_date_value && get_value(orders[i]) <= upper_date_value){
            // Guarda las órdenes dentro del rango en el vector de resultados
            results.push_back(orders[i]);
        }
    }

    // Imprime los resultados de la búsqueda y los escribe en el archivo de salida
    cout << "Resultados de la búsqueda:" << endl;
    search_results << "Resultados de la búsqueda \"" << lower_date << " - " << upper_date << "\"" << endl;

    if(results.size() == 0){
        cout << "No hay resultados para esta fecha";
    }

    else{
        for(j = 0; j < results.size(); j++){
            cout << j + 1 << "- " << results[j] << endl;
            search_results << j + 1 << "- " << results[j] << endl;
        }
    }

    search_results.close();
}

```

En la primera, se realiza la búsqueda en una fecha en específico y los parámetros de entrada son el string de la fecha (proporcionado por el usuario a través de la consola), su valor entero (obtenido a través de “get_value”), el vector que contiene las órdenes (ya ordenado) y el tamaño del mismo. En la segunda, se hace una búsqueda dentro de un rango de fechas, se reciben string de las fechas límite (dadas por el usuario), su valor entero (obtenido con “get_value”), el vector con las órdenes (ya ordenado) y su tamaño. En ambos casos, se hace uso de la búsqueda secuencial, la cual tiene una complejidad de $O(n)$ y es extremadamente sencilla de implementar y adaptar a los requerimientos del proyecto. Los resultados de la búsqueda son almacenados en un vector, se despliegan en la consola y son guardados en un archivo de texto para que el usuario pueda acceder a estos en cualquier momento.

Para mantener un entorno de desarrollo organizado, las funciones de ordenamiento, búsqueda y obtención de valores están almacenadas en archivos header (.h) y se importan dentro del código principal.

Pruebas

Ordenamiento

```
PS C:\Users\bruno\OneDrive\Documentos\situacion problema> ./main.exe orders.txt
Primeras 10 ordenes
ene 1 12:36:28 R:La Cocina del Chef O:Sopa de Tomate(130)
ene 1 13:0:36 R:La Casa del Sabor O:Lasana Vegetariana(246)
ene 1 13:28:12 R:The elegant eatery O:Bratwurst con Chucrut(307)
ene 1 13:40:23 R:The elegant eatery O:Paella de Mariscos(457)
ene 1 13:49:3 R:La Furia O:Pollo con Mole(276)
ene 1 13:54:31 R:La Casa del Sabor O:Filete de Res(373)
ene 1 14:20:22 R:Los Rincones O:Cazuela de Mariscos(376)
ene 1 14:27:14 R:El Cafe del Valle O:Pollo al Curry(277)
ene 1 14:47:53 R:The Trendy Cafe O:Tiramisu(141)
ene 1 15:28:38 R:El Sol O:Pad See ew(270)
```

```
main.cpp sorted_orders.txt X
outputs > sorted_orders.txt
1 ene 1 12:36:28 R:La Cocina del Chef O:Sopa de Tomate(130)
2 ene 1 13:0:36 R:La Casa del Sabor O:Lasana Vegetariana(246)
3 ene 1 13:28:12 R:The elegant eatery O:Bratwurst con Chucrut(307)
4 ene 1 13:40:23 R:The elegant eatery O:Paella de Mariscos(457)
5 ene 1 13:49:3 R:La Furia O:Pollo con Mole(276)
6 ene 1 13:54:31 R:La Casa del Sabor O:Filete de Res(373)
7 ene 1 14:20:22 R:Los Rincones O:Cazuela de Mariscos(376)
8 ene 1 14:27:14 R:El Cafe del Valle O:Pollo al Curry(277)
9 ene 1 14:47:53 R:The Trendy Cafe O:Tiramisu(141)
10 ene 1 15:28:38 R:El Sol O:Pad See ew(270)
11 ene 1 15:35:34 R:El Cafe del Chef Pablo O:Hamburguesa Gourmet(280)
12 ene 1 16:8:37 R:The Bistro Lounge O:Pad See ew(291)
13 ene 1 16:15:56 R:The elegant eatery O:Ratatouille(306)
14 ene 1 16:31:16 R:The Classic O:Tortilla espanola(300)
15 ene 1 18:0:48 R:El Mercado del Pueblo O:Fondue de Queso(410)
16 ene 1 18:15:14 R:The Bee O:Chop Suey(257)
17 ene 1 18:15:42 R:The Urban O:Salmon Gravlax(295)
18 ene 1 18:49:44 R:Por la Playa O:Chop Suey(257)
19 ene 1 18:53:8 R:Ondina O:Bratwurst con Chucrut(277)
20 ene 1 18:54:28 R:La esquina del Chef O:Bistec a la Pimienta(273)
21 ene 1 19:6:47 R:El Jardin de los Sabores O:Ceviche Mixto(275)
22 ene 1 19:8:6 R:La Cabana O:Spaghetti Carbonara(252)
23 ene 1 20:48:35 R:The Trendy eatery O:Beef Wellington(445)
24 ene 1 21:32:57 R:The Corner Place O:Carne Asada(290)
25 ene 2 12:28:16 R:The Trendy Table O:Fondue de Queso(436)
26 ene 2 12:39:10 R:The Gourmet Lala O:Shawarma de Pollo(284)
27 ene 2 13:14:14 R:El Mar O:Sashimi de Atun(261)
28 ene 2 14:1:56 R:El dragon O:Sopa de Tortilla(134)
29 ene 2 15:58:58 R:Urban eats O:Sashimi de Atun(311)
30 ene 2 17:30:9 R:El Cafe de Lola O:Cordero al Romero(444)
31 ene 2 17:32:9 R:Sabor a Mar O:Huevos Benedictinos(268)
32 ene 2 17:48:50 R:El Cafe del Valle O:Risotto de Champinones(407)
33 ene 2 18:4:24 R:El Pueblo O:Churrasco(413)
34 ene 2 18:35:6 R:La Internacional O:Kebab de Cordero(281)
35 ene 2 20:47:6 R:The Modern eatery O:Curry de Pollo(268)
36 ene 2 21:34:36 R:La Comida Casera O:Ribeye Steak(462)
37 ene 2 23:39:0 R:La Terraza del Mar O:Beef Wellington(438)
```



```
main.cpp | sorted_orders.txt X
outputs > sorted_orders.txt
8420 Nov 26 13:59:50 R:The Artisan Table O:Ribeeye Steak(390)
8421 Nov 26 14:7:27 R:Tierra y Mar O:Ceviche de Camaron(269)
8422 Nov 26 15:0:1 R:La Terraza del Mar O:Carne Asada(292)
8423 Nov 26 15:25:24 R:The Bee O:Pizza Margherita(255)
8424 Nov 26 15:30:33 R:The Rustic Spoon O:Beef Stroganoff(420)
8425 Nov 26 15:49:12 R:Harbor Lights Restaurant O:Mousse de Chocolate(159)
8426 Nov 26 16:3:57 R:The Cafe Lounge O:Fish and Chips(270)
8427 Nov 26 16:24:0 R:The Bistro at Sunset O:Filete Mignon(396)
8428 Nov 26 16:28:47 R:La Bella Notte O:Tortilla espanola(275)
8429 Nov 26 16:32:56 R:The Gourmet Deli O:Cochinita Pibil(267)
8430 Nov 26 16:34:30 R:El Sol O:Tacos al Pastor(137)
8431 Nov 26 16:38:53 R:The Urban Bistro O:Burrata con Tomate(285)
8432 Nov 26 16:43:29 R:Ocean O:enchiladas Verdes(252)
8433 Nov 26 17:7:3 R:La Cocina del Mercado O:Burrata con Tomate(285)
8434 Nov 26 17:11:9 R:La Cantina de Juan O:Pasta Alfredo(285)
8435 Nov 26 17:41:2 R:El Asador de la Abuela O:Bistec a la Pimienta(280)
8436 Nov 26 18:4:0 R:The elegant Bistro O:escargots a la Bourguignonne(318)
8437 Nov 26 18:15:50 R:La Bella Notte O:Fideua(279)
8438 Nov 26 18:35:33 R:The elegant Bistro O:Pollo Teriyaki(300)
8439 Nov 26 19:8:24 R:El Sabor del Pueblo O:Dim Sum(267)
8440 Nov 26 19:8:40 R:Bosque Real O:ensalada Griega(143)
8441 Nov 26 19:19:34 R:The Chic Bistro O:Boeuf Bourguignon(406)
8442 Nov 26 19:31:3 R:La Trattoria Bella O:escargots a la Bourguignonne(269)
8443 Nov 26 20:27:50 R:The eatery O:Boeuf Bourguignon(412)
8444 Nov 26 20:43:38 R:Ondina O:Bacalao a la Vizcaina(412)
8445 Nov 26 20:55:42 R:The Modern eatery O:Tiradito de Pescado(265)
8446 Nov 26 20:57:54 R:El Sazon Mexicano O:Burrata con Tomate(270)
8447 Nov 26 21:22:45 R:The Seafood Shack O:Arroz con Leche(136)
8448 Nov 26 23:13:58 R:Don Perfecto O:Salmon al Horno(278)
8449 Nov 27 13:15:13 R:The Garden Table O:Salmon al Horno(284)
8450 Nov 27 13:18:9 R:El Cafe de la Abuela O:Pasta con Mariscos(389)
8451 Nov 27 13:20:19 R:La Furia O:Osso Buco(416)
8452 Nov 27 14:6:33 R:The Waterfront O:Croquetas de Jamon(156)
8453 Nov 27 14:38:54 R:The Table O:Tacos al Pastor(137)
8454 Nov 27 14:56:9 R:Here and Now O:Cordero al Romero(431)
8455 Nov 27 15:21:58 R:The Farm-to-Table O:Bistec a la Pimienta(319)
8456 Nov 27 15:39:14 R:La Cabana O:Moussaka(253)
```

Búsqueda por fecha y hora

Fecha y hora específica

```
-----
Busqueda de ordenes por fecha y hora
-----
Seleccione una opcion:
1- Busqueda individual
2- Busqueda por rango
> 1

Ingrese la fecha y hora a buscar (ej. "Jun 12 06:51:16"):
> Jun 24 19:39:45

Resultados de la busqueda:
1- Jun 24 19:39:45 R:La Terraza del Mar O:Falafel con Hummus(284)
```

```
main.cpp | search_results.txt M X
outputs > search_results.txt
1 Resultados de la busqueda "Jun 24 19:39:45"
2 1- Jun 24 19:39:45 R:La Terraza del Mar O:Falafel con Hummus(284)
3
```

Rango de fechas

```
-----
Busqueda de ordenes por fecha y hora
-----
Seleccione una opcion:
1- Busqueda individual
2- Busqueda por rango
> 2

Ingrese las fechas y horas (ej. "Jun 12 06:51:16"):
Limite inferior: Jun 12 00:00:00
Limite superior: Jun 12 23:59:59
Resultados de la busqueda:
1- Jun 12 12:38:3 R:The Rustic Spoon O:Tacos de Lengua(279)
2- Jun 12 12:52:19 R:La Francesa O:Fajitas de Res(264)
3- Jun 12 13:9:48 R:The Local Bistro O:Pollo Tikka Masala(281)
4- Jun 12 13:10:5 R:La Bella Notte O:Coq au Vin(421)
5- Jun 12 13:52:24 R:La Cocina de Pablo O:Tacos de Birria(273)
6- Jun 12 14:49:21 R:The Modern Palate O:Ceviche Mixto(279)
7- Jun 12 14:57:53 R:The Farmhouse Cafe O:Arroz Frito(287)
8- Jun 12 14:59:43 R:El Cafe de la Abuela O:Tacos al Pastor(132)
9- Jun 12 15:9:19 R:El Cafe de la Abuela O:Crepas con Nutella(131)
10- Jun 12 15:12:59 R:The Stylish Diner O:Ceviche Mixto(292)
11- Jun 12 15:27:11 R:El Cafe de la Plaza O:Kebab de Cordero(273)
12- Jun 12 15:29:40 R:Dos hermanos O:ensalada Cesar(298)
13- Jun 12 15:34:51 R:La Cocina del Chef O:Panqueques con Miel(130)
14- Jun 12 16:4:8 R:The Seafood Shack O:Moussaka(277)
15- Jun 12 16:21:10 R:Las salsas O:Sopa de Cebolla(151)
16- Jun 12 16:26:51 R:Rios O:empanadas de Carne(151)
17- Jun 12 16:41:20 R:Las salsas O:Costillas BBQ(445)
18- Jun 12 16:53:12 R:Urban eats O:Chop Suey(308)
19- Jun 12 17:28:16 R:The elegant Kitchen O:Ribeye Steak(413)
20- Jun 12 17:43:46 R:Bistro Boulevard O:Pollo al Limon(294)
21- Jun 12 18:2:18 R:Classic Cafe O:Arroz Frito(291)
22- Jun 12 18:14:33 R:The Green Room O:Arrachera Asada(316)
23- Jun 12 18:28:55 R:El Otro Rincon O:Chimichurri Steak(410)
24- Jun 12 18:38:38 R:El Uro O:Raviolis de espinaca(280)
25- Jun 12 18:42:58 R:The Lounge O:Ratatouille(283)
26- Jun 12 18:46:3 R:El Restaurante de Ana O:Arroz con Leche(137)
27- Jun 12 18:48:57 R:La esquina del Taco O:ensalada Griega(129)
28- Jun 12 19:0:4 R:La Cocina de la Abuela O:Burrata con Tomate(272)
29- Jun 12 19:3:51 R:The Stylish Diner O:Filete Mignon(439)
30- Jun 12 19:32:7 R:El Grande O:Fajitas de Res(296)
31- Jun 12 19:46:12 R:El dragon O:Pozole Blanco(136)
32- Jun 12 19:52:8 R:City Bistro O:Sushi Roll(301)
33- Jun 12 20:3:43 R:The Savory O:Cochinita Pibil(276)
34- Jun 12 20:5:26 R:The Bee O:ensalada Caprese(131)
35- Jun 12 20:31:24 R:La Pasticceria O:Curry de Garbanzos(308)
36- Jun 12 20:36:38 R:El Mar O:Mole Poblano(261)
37- Jun 12 21:33:10 R:El Rincon del Chef O:Pollo Tikka Masala(296)
38- Jun 12 23:0:2 R:Urban eats O:Crema de Champinones(158)
```

```
main.cpp M search_results.txt M X
outputs > search_results.txt
1 Resultados de la busqueda "Jun 12 00:00:00 - Jun 12 23:59:59"
2 1- Jun 12 12:38:3 R:The Rustic Spoon 0:Tacos de Lengua(279)
3 2- Jun 12 12:52:19 R:La Francesa 0:Fajitas de Res(264)
4 3- Jun 12 13:9:48 R:The Local Bistro 0:Pollo Tikka Masala(281)
5 4- Jun 12 13:10:5 R:La Bella Notte 0:Coq au Vin(421)
6 5- Jun 12 13:52:24 R:La Cocina de Pablo 0:Tacos de Birria(273)
7 6- Jun 12 14:49:21 R:The Modern Palate 0:Ceviche Mixto(279)
8 7- Jun 12 14:57:53 R:The Farmhouse Cafe 0:Arroz Frito(287)
9 8- Jun 12 14:59:43 R:El Cafe de la Abuela 0:Tacos al Pastor(132)
10 9- Jun 12 15:9:19 R:El Cafe de la Abuela 0:Crepas con Nutella(131)
11 10- Jun 12 15:12:59 R:The Stylish Diner 0:Ceviche Mixto(292)
12 11- Jun 12 15:27:11 R:El Cafe de la Plaza 0:Kebab de Cordero(273)
13 12- Jun 12 15:29:40 R:Dos hermanos 0:ensalada Cesar(298)
14 13- Jun 12 15:34:51 R:La Cocina del Chef 0:Panqueques con Miel(130)
15 14- Jun 12 16:4:8 R:The Seafood Shack 0:Moussaka(277)
16 15- Jun 12 16:21:10 R:Las salsas 0:Sopa de Cebolla(151)
17 16- Jun 12 16:26:51 R:Rios 0:empanadas de Carne(151)
18 17- Jun 12 16:41:20 R:Las salsas 0:Costillas BBQ(445)
19 18- Jun 12 16:53:12 R:Urban eats 0:Chop Suey(308)
20 19- Jun 12 17:28:16 R:The elegant Kitchen 0:Ribeye Steak(413)
21 20- Jun 12 17:43:46 R:Bistro Boulevard 0:Pollo al Limon(294)
22 21- Jun 12 18:2:18 R:Classic Cafe 0:Arroz Frito(291)
23 22- Jun 12 18:14:33 R:The Green Room 0:Arrachera Asada(316)
24 23- Jun 12 18:28:55 R:El Otro Rincon 0:Chimichurri Steak(410)
25 24- Jun 12 18:38:38 R:El Uro 0:Raviolis de espinaca(280)
26 25- Jun 12 18:42:58 R:The Lounge 0:Ratatouille(283)
27 26- Jun 12 18:46:3 R:El Restaurante de Ana 0:Arroz con Leche(137)
28 27- Jun 12 18:48:57 R:La esquina del Taco 0:ensalada Griega(129)
29 28- Jun 12 19:0:4 R:La Cocina de la Abuela 0:Burrata con Tomate(272)
30 29- Jun 12 19:3:51 R:The Stylish Diner 0:Filete Mignon(439)
31 30- Jun 12 19:32:7 R:El Grande 0:Fajitas de Res(296)
32 31- Jun 12 19:46:12 R:El dragon 0:Pozole Blanco(136)
33 32- Jun 12 19:52:8 R:City Bistro 0:Sushi Roll(301)
34 33- Jun 12 20:3:43 R:The Savory 0:Cochinita Pibil(276)
35 34- Jun 12 20:5:26 R:The Bee 0:ensalada Caprese(131)
36 35- Jun 12 20:31:24 R:La Pasticceria 0:Curry de Garbanzos(308)
37 36- Jun 12 20:36:38 R:El Mar 0:Mole Poblano(261)
38 37- Jun 12 21:33:10 R:El Rincon del Chef 0:Pollo Tikka Masala(296)
39 38- Jun 12 23:0:2 R:Urban eats 0:Crema de Champinones(158)
40
```

Referencias

Findus Foodservice & <https://www.findusfoodservices.es/humans.txt>. (s. f.). *¿Qué es el*

Delivery Food? Findus Foodservices.

<https://www.findusfoodservices.es/novedades/que-es-el-delivery-food/>