



11 DE DICIEMBRE DE 2021

# PRÁCTICA FINAL: INGESTA CALÓRICA

INTERFACES GRÁFICAS DE USUARIO. 3º GII

MARÍA GONZÁLEZ GARCÍA  
UNIVERSIDAD DE SALAMANCA  
2021-2022

---

## ÍNDICE

---

<b>1. MANUAL DE USUARIO</b>	<b>2</b>
<b>1.1. VENTANA PRINCIPAL</b>	<b>2</b>
1.1.1. MENÚ DESPLEGABLE	2
1.1.2. VISTA DE INICIO	3
1.1.3. VISTA COMPARATIVA ENTRE DÍAS	3
1.1.4. VISTA COMPARATIVA ENTRE COMIDAS	4
1.1.5. VISTA COMPARATIVA CON LA MEDIA RECOMENDABLE	5
<b>1.2. VENTANA DE DISEÑO</b>	<b>5</b>
<b>1.3. VENTANA DE DATOS</b>	<b>6</b>
<b>1.4. PUNTOS DE INTERÉS</b>	<b>7</b>
<b>2. MANUAL DEL PROGRAMADOR</b>	<b>9</b>
<b>2.1. DIAGRAMA DE CLASES</b>	<b>9</b>
<b>2.2. MAINWINDOW</b>	<b>9</b>
<b>2.3. TABLADATOS</b>	<b>10</b>
<b>2.4. TABLAEVENTARGS</b>	<b>11</b>
<b>2.5. BARRASEVENTARGS</b>	<b>11</b>
<b>2.6. COLOREVENTARGS</b>	<b>11</b>
<b>2.7. VENTANADISEÑO</b>	<b>11</b>
<b>2.8. CALORIASDIAS</b>	<b>12</b>
<b>2.9. TABLACOMIDAS</b>	<b>12</b>
<b>3. BIBLIOGRAFÍA</b>	<b>13</b>

## 1. MANUAL DE USUARIO

La aplicación, llamada “Ingesta Calórica”, se ha desarrollado para resolver el problema propuesto, que se basaba en la representación, a través de un gráfico de barras, de la ingesta calórica del usuario en distintas ocasiones. De esta manera, se presenta una comparativa de barras tanto para una serie de días como para las comidas concretas de un día específico.

Esta aplicación nos muestra varias ventanas, cada una de ellas con una funcionalidad distinta, consiguiendo así alcanzar el objetivo final y conjunto de la práctica. Las ventanas y vistas son las siguientes:

### 1.1. Ventana principal

La ventana principal siempre debe permanecer siempre abierta, ya que, si se cerrase, se saldría de la aplicación y se perderían los datos. En ella podremos observar una página de inicio que da lugar a una vista general del gráfico de barras, representando cada una de ellas un día distinto de los introducidos y divide en tantos fragmentos como comidas se hayan consumido; así como también un gráfico que compara las comidas de un mismo día, habiendo tenido este que ser previamente seleccionado en otra ventana, y una vista de las calorías medias ingeridas en un día en relación con la media de las recomendadas por especialistas.

Todas las distintas vistas cuentan con algo en común: un menú que nos permite desplegar algunas opciones para realizar distintas funciones.

#### 1.1.1. Menú desplegable

Se compone por dos opciones (a los que podemos acceder pulsando la tecla *Alt* de nuestro ordenador seguida del carácter subrayado una vez la pulsemos).

El primero, “*Archivo*”, se compone de dos posibles opciones:

- **Exportar gráfico**: que nos permite exportar todo aquello que se esté mostrando en el espacio de dibujo en ese momento como una imagen.
- **Salir**: que mostrará un cuadro de diálogo preguntándonos si deseamos salir de la aplicación y esperará nuestra confirmación o cancelación.

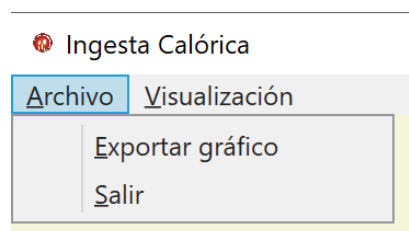


Ilustración 1. Menú “Archivo”

El segundo desplegable es el de “*Visualización*”, que nos permitirá acceder a las demás ventanas:

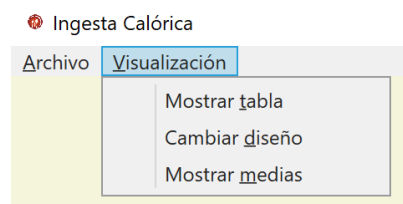


Ilustración 2. Menú "Visualización"

- **Mostrar tabla de datos**: nos permitirá abrir la [ventana dedicada a la introducción y presentación de los datos](#).
- **Cambiar diseño**: nos dará la posibilidad de abrir la [ventana de diseño](#) y elegir un color entre algunos disponibles para poder cambiar automáticamente el color de la ventana principal.
- **Mostrar medias**: nos enseñará una nueva vista de la ventana principal donde se mostrarán las [medias de las calorías de los días](#) con información asociada en relación con las medias recomendadas por los especialistas.

### 1.1.2. Vista de inicio

Nos da la bienvenida a la aplicación. En ella encontramos el logo de esta (también visible como icono en todas las ventanas que abramos), una frase que hace referencia al objetivo fundamental del programa y un botón, en el que si pulsamos seremos dirigidos a la [vista general](#), en un primer momento vacía, donde se mostrará la comparativa de las calorías ingeridas en los distintos días introducidos.

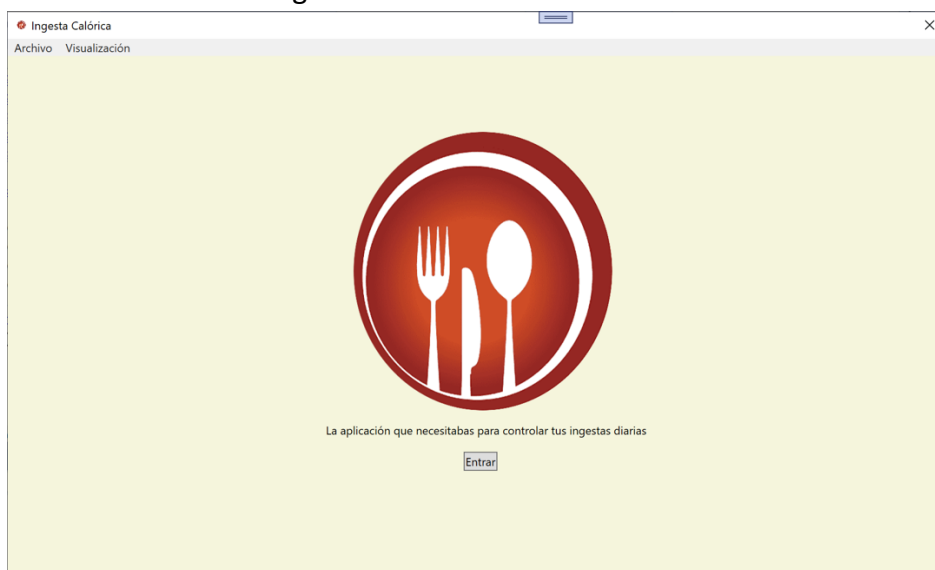


Ilustración 3. Página de inicio

### 1.1.3. Vista comparativa entre días

Es a donde se nos dirige una vez hemos pulsado el botón de la portada. En ella, se mostrarán 23 barras, como máximo, de la colección que hemos estado tratando (donde hemos estado introduciendo los datos), añadiéndose y modificándose las líneas según se vayan agregando más.

Estas barras estarán divididas por colores, cada una representando una comida del día distinta, y su altura será la suma de las calorías ingeridas en cada una de las comidas.

De esta manera, podremos observar una evolución del consumo del usuario a lo largo de los días.

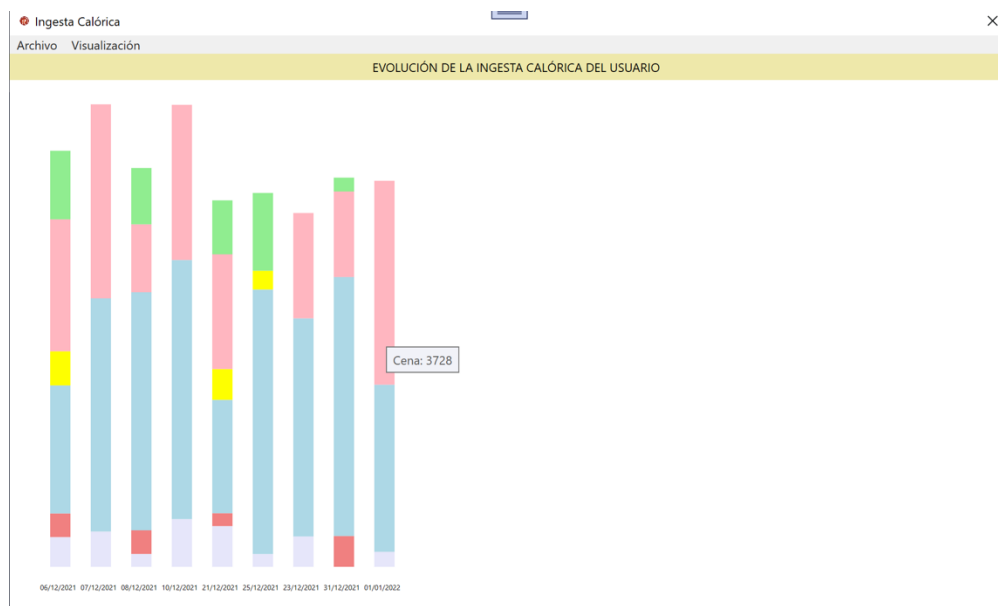


Ilustración 4. Vista general de los datos

#### 1.1.4. Vista comparativa entre comidas

Cuando seleccionemos un día concreto en la [tabla](#) que veremos en uno de los siguientes apartados (que cuenta con las fechas de los días que tienen datos asociados y el total de las calorías consumidas), podremos acceder esta vista, la cual nos hace una comparativa entre el consumo de calorías por parte del usuario en las distintas comidas que haya ingerido un día concreto (el seleccionado).

Cuenta con un botón que nos permite volver a la vista explicada [en el punto anterior](#) siempre que el usuario lo desee.



Ilustración 5. Vista del gráfico de un día concreto

#### 1.1.5. Vista comparativa con la media recomendable

Cuando seleccionemos esta opción ([en el menú de Visualización](#)), se nos cambiará la vista de la ventana principal. Se mostrará entonces un nuevo gráfico que contará de tantos puntos como días que recojan información en la lista, y tres líneas, cada una de ellas representando el valor medio recomendable de calorías ingeridas en distintos casos.

Cada punto corresponde al valor medio de las calorías ingeridas en un día (el total entre las seis distintas opciones de comidas que se ofrecen).

En el caso de las barras, la azul corresponde a la media recomendada para los hombres; la rosa, para las mujeres y, la negra, el consumo general.

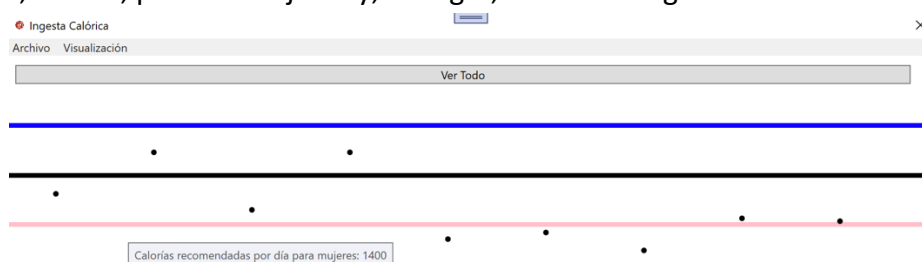


Ilustración 6. Vista del gráfico de medias

#### 1.2. Ventana de diseño

Esta ventana nos permite realizar la función de cambio de color del fondo de pantalla descrita en [este apartado](#).

Cuando accedamos a ella a través del menú desplegable, se abrirá y, en ella, podremos encontrar diversos botones de distintos colores que, al pulsarlos, harán que el fondo de la zona de dibujo de nuestra ventana principal adquiera la tonalidad del botón pulsado. Su actualización es automática.



*Ilustración 7. Aspecto de la ventana de diseño*



*Ilustración 8. Vista de la ventana principal una vez pulsado uno de los botones de la Ilustración 7*

### 1.3. Ventana de datos

Es la dedicada a la introducción de datos y su presentación en tablas, y a ella accedemos desde una de las opciones del [menú de Visualización](#).

En esta ventana encontramos varios objetos de interés:





3. El usuario debe contar con que, como ya se ha dicho, no debe cerrar la ventana principal (titulada *Ingesta Calórica*) si desea introducir más datos, ya que su cierre conlleva el cierre total de la aplicación.
4. Finalmente, cabe destacar que, en el caso de los gráficos, todas las formas agregadas tienen un *Tooltip* asociado. Esto se puede apreciar en algunas de las ilustraciones anteriores (4, 5 y 6, para ser más específicos):
  - En el caso de la [vista general \(comparativa entre días\)](#), cuando posamos el cursor sobre alguno de los tramos de la línea, se nos especificará tanto la comida a la que pertenece dicho fragmento como las calorías ingeridas. También, si nos colocamos sobre la fecha, se nos indicará la cantidad total consumida dicho día.
  - En el caso de la [vista comparativa entre comidas](#), al posarnos sobre las barras, se nos indicará su valor asociado.
  - En el caso de la [vista comparativa con la media recomendable](#), al posarnos sobre los puntos, se nos indica el día y la media de calorías; y al posarnos sobre las barras, se nos indicará cuál es su ámbito de estudio (mujeres/hombres/general) y el valor asociado.

## 2. MANUAL DEL PROGRAMADOR

### 2.1. Diagrama de clases

Empezamos el manual mostrando una relación entre las clases usadas en nuestro proyecto:

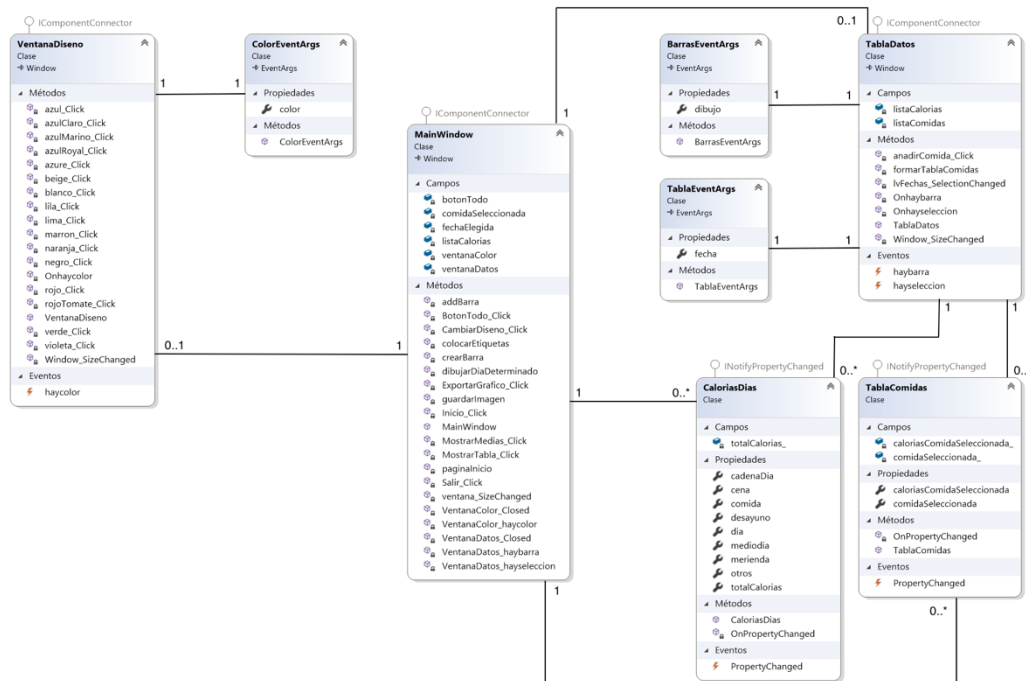


Ilustración 10. Diagrama de clases del proyecto

### 2.2. MainWindow

Contiene las vistas y elementos explicados en [este punto](#). Posee algunas variables globales: dos para instanciar cada una de las otras dos ventanas (una de cada tipo), el botón que se muestra en la parte superior de [esta vista](#), un *DateTime* que se utiliza cuando se ha seleccionado una fecha en el *ListView* para poder mostrar [este gráfico](#), y dos listas (una que guarda toda la información de todos los días y otra que se usa para almacenar la información de un día específico).

Esta ventana tiene muchos métodos (algunos, como los *Click* de los botones o la creación de una nueva ventana, son tan simples que no se explicarán, ya que su comportamiento es básico – si no está instanciado el objeto en sí, se instanciará, y se le dotará de las propiedades convenientes). Algunos métodos más complejos o destacables son los siguientes:

- **`private void paginalnicio()`**: Se encarga de crear [la vista de bienvenida](#) (una etiqueta con la frase, un botón que nos permite acceder a la aplicación y un rectángulo cuyo relleno es una imagen, el logo usado para la aplicación).
- **`private void Salir_Click(object sender, RoutedEventArgs e)`**: se dispara cuando damos a la opción *Salir* de [este menú](#), mostrándonos un cuadro de diálogo en modo modal que

nos preguntará si deseamos salir o no de la aplicación, cerrando todas las ventanas en caso de confirmación.

- **private void addBarra()**: se encarga de crear las barras que se muestran en el [gráfico comparativo entre días](#), recorriendo la lista hasta que se alcance el máximo de estas permitido para mostrar en la zona de dibujo (un canvas en este caso).
- **private void dibujarDiaDeterminado(DateTime fecha)**: se encarga de crear la vista del [gráfico comparativo entre comidas](#), llamando a otras funciones para mostrar el resultado final.
- **private void crearBarra(int max, int cal, int coordX, Brush color)**: función a la que recurre la anterior y que se encarga de crear las barras de dicho gráfico.
- **private void colocarEtiquetas()**: función llamada por *dibujarDiaDeterminado* que se encarga de colocar las etiquetas que nos indican qué día estamos visualizando y las del tipo de comida encontradas al pie de canvas.
- **private void VentanaDatos haybarra(object sender, BarrasEventArgs e)**: hace referencia al delegado del evento que nos indica que hay barras de la [vista general](#) que dibujar al introducir un nuevo dato.
- **private void VentanaDatos hayseleccion(object sender, TablaEventArgs e)**: referencia el delegado del evento que indica que una fecha ha sido seleccionada y debemos preparar [la vista de ese día](#).
- **private void MostrarMedias\_Click(object sender, RoutedEventArgs e)**: se dispara cuando se clic en la acción correspondiente del [menú desplegable](#) para poder mostrar el [gráfico comparativo entre medias](#).
- **private void ExportarGrafico\_Click(object sender, RoutedEventArgs e)**: muestra un cuadro de diálogo en modo modal cuando deseamos [exportar nuestro gráfico](#).
- **private void guardarImagen(string fileName)**: crea la imagen a exportar del método anterior.
- **private void ventana SizeChanged(object sender, SizeChangedEventArgs e)**: se encarga de determinar que la ventana no sea redimensionable para evitar fallos.

### 2.3. TablaDatos

Posee dos listas (que será equivalentes a las dos de MainWindow) y dos delegados para los eventos asociados a las siguientes dos clases, así como también dos métodos para disparar cada uno (*Onhayseleccion* pasa la fecha determinada cuando se selecciona un ítem en el *ListView*, y *Onhaybarra*, que pasa un valor booleano para indicar que se debe pintar una línea al introducir nuevos datos o modificar los existentes), así como su constructor (que recibe las dos listas mencionadas anteriormente de la ventana principal) y el método que establece que la ventana no es redimensionable. Algunos métodos más son:

- **private void anadirComida\_Click(object sender, RoutedEventArgs e):** se encarga de añadir a la lista de todos los datos los nuevos cuando clicamos el botón. Desde este método, se llama también a *Onhaybarra* para proceder a dibujar el nuevo dato.
- **private void formarTablaComidas():** invocado por la función anterior y por el método de cambio de selección en la lista de las fechas, se encarga de introducir los datos correspondientes a ese día en la lista de las comidas y actualizarlos en caso de ser necesario.
- **private void lvFechas\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e):** informa a la ventana principal (usando *Onhayseleccion*) y al método anterior de que un elemento del *ListView* de fechas se ha seleccionado para que se proceda de la manera adecuada.

## 2.4. TablaEventArgs

Se trata de una clase encargada de notificar que se ha seleccionado un elemento del *ListView* de las fechas y transmitirle esta información a la ventana principal para que se dibuje el [gráfico comparativo entre comidas](#) de un día concreto. Se compone por una variable de tipo *DateTime* que recoge la fecha del ítem seleccionado y un constructor que le da un valor inicial.

## 2.5. BarrasEventArgs

Se trata de una clase encargada de notificar a la ventana principal que se han añadido datos nuevos a la colección y que es necesario redibujar el [gráfico comparativo entre días](#) para mostrarlos al usuario. Se compone por una variable booleana que transmitirá esa información y un constructor (al igual que la clase anterior).

## 2.6. ColorEventArgs

Se trata de una clase encargada de notificar a la principal que se ha decidido cambiar su color de fondo, componiéndose por una variable que lo recoge (de tipo *Brush*) y su correspondiente constructor (al igual que las dos clases anteriores).

## 2.7. VentanaDiseno

Es una clase simple que se encarga del cambio de color de fondo de la ventana. Se compone por un método que dispara el evento que hace referencia a la clase [ColorEventArgs](#) (*Onhaycolor*, que funciona igual que los ya explicados), uno que impide la redimensión de la ventana (*Window\_SizeChanged*) y muchos métodos *Click*, uno por cada botón de cambio de color que se muestra, cuya forma es la siguiente:

- **private void nombreBoton\_Click(object sender, RoutedEventArgs e):** se encarga de invocar a *Onhaycolor* y pasarle el color de relleno de cada botón, permitiendo el cambio automático del fondo de la ventana principal (el canvas y el panel donde colocamos el botón, para ser más exactos).

## 2.8. CaloriasDias

Esta clase nos permite almacenar todos los datos pertenecientes a un día concreto, desde la fecha seleccionada en el [calendario de la ventana de datos](#) hasta su versión como cadena (que recoge solo el día, mes y año), pasando por cada uno de los datos calóricos de cada comida y la suma de todos ellos. Cuenta con un constructor que les da valores por defecto (0 en caso de las calorías, la fecha del día actual en la variable que recogerá la seleccionada por el calendario y una cadena vacía en la restante).

Además, esta clase destaca porque tiene implementada la interfaz *INotifyPropertyChanged*, relacionada a la variable del total de calorías; así como, por consecuencia, el delegado del evento correspondiente, *PropertyChanged*, y el método que lo dispara *OnPropertyChanged*, que es activado cuando ese dato varía, haciendo que este hecho sea visible para el resto de clases y que, por ejemplo, se pueda actualizar el *ListView* que recoge las fechas con su total de calorías consumidas.

## 2.9. TablaComidas

Su implementación es bastante similar a la de la clase anterior, por lo que no necesita mucha explicación. Simplemente indicar que, en este caso, la interfaz *INotifyPropertyChanged* se implementa para las dos variables que recoge (una cadena que indica el nombre de la comida a mostrar y un entero con las calorías consumidas en dicha comida). Esto nos permite que, cuando se seleccione un ítem en la tabla de las fechas, se cree una instancia de esta clase y que vaya recorriendo los datos de ese día concreto. Al cambiar el valor de las variables de esta clase, se dispara el método del evento de la interfaz (*OnPropertyChanged*) y se irán añadiendo a una lista que se usará como fuente para el *ListView* que detalla las comidas específicas del día seleccionado.

### 3. BIBLIOGRAFÍA

A continuación, se presenta una lista de las páginas o manuales consultados para darle solución al enunciado propuesto:

- [Información general sobre alineación, márgenes y relleno](#)
- [Información general sobre la clase Debug](#)
- [Información general sobre la clase RenderTargetBitmap](#)
- [Guardar canvas como imagen](#)
- [Guardar canvas como imagen 2](#)
- [Visibilidad de etiquetas](#)
- [Situar elementos en el canvas](#)
- [Eliminación de filas en ListView](#)
- [Gráficos en WPF](#)
- [BarChart en C#](#)
- [Cambiar formato de DateTime](#)
- [Actualización de ListView](#)
- [UpdateItem de ListView](#)
- [SelectedValue de ListView](#)
- [Información sobre delegados](#)
- [Variable globales](#)
- [Variables globales 2](#)
- [Inserción de Canvas en ScrollViewer](#)
- [Información general sobre ScrollViewer](#)
- [Instalación del diseñador de clases](#)
- Apuntes de la asignatura.
- Códigos de soluciones que se subieron al curso de Studium.
- Códigos de las prácticas hechas en clase.