

Actividad | #3 | Ejecución

Lenguajes de Programación II

Ingeniería en Desarrollo de Software

TUTOR: Félix Acosta Hernández

ALUMNO: Alejandra Ibarra Carmona

FECHA: 08/12/2024

Indicé

Introducción…………………………………………..3

Descripción…………………………………………..3

Justificación………………………………………….3

Desarrollo……………………………………………4

○ Ejecución………………………………………….4

○ Código…………………………………………….4

Conclusión…………………………………………7

Referencias…………………………………………7

Introducción

Dentro de esta actividad se estará mostrando nuevamente con la conexión a nuestra base de datos la tabla, pero ya con los valores correspondientes que se introdujeron al inicio en la actividad uno.

En C++ los miembros de la clase pueden llegar a tener tres de las siguientes modificaciones de acceso público, privado o protegido, unas de las mejores practicas sugieren que se ordenen los componentes de las clases de acuerdo con sus modificaciones.

Un caso particular de protected, es considerado un punto intermedio entre public, y private, cada uno de los miembros clasificados en protected se comportan como public para todos los métodos de la clase derivada y amigas y como private para el resto de las funciones.

Descripción

El encapsulamiento es una tecnica que permite localizar y ocultar los detalles de un objeto, el encapsulamiento previene que un objeto sea manipulado por operaciones distintas y definidas. La encapsulación es como una caja negra que esconde los datos y solamente permite acceder a ellos de forma controlada, existen un par de razones técnicas para la utilización en la encapsulación:

* Mantener a salvo los detalles de representación si solo estamos enfocados en el comportamiento del objeto
* Modificar y ajustar la representación a mejores soluciones algorítmicas o nuevas tecnologías del software

La encapsulación define los niveles de acceso para elementos de determinada clase, estos niveles definen el derecho de acceso para los datos, los métodos de get y set usualmente definen para el acceso de una variable privada de la clase, para obtener su estado actual es decir su valor y modificarlo respectivamente.

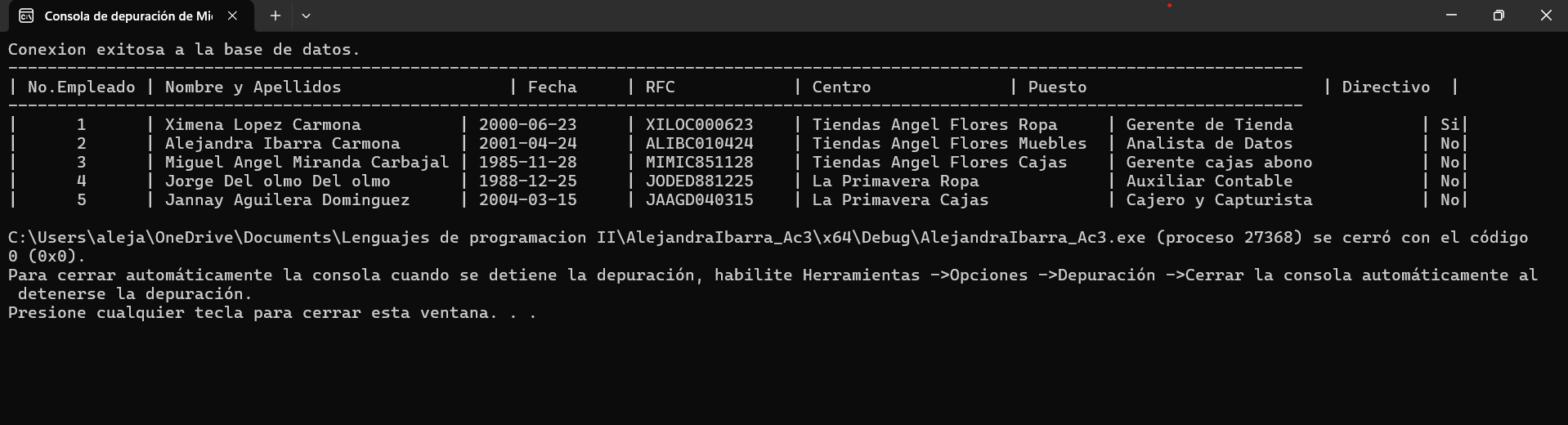
Justificación

Un método get consta por lo general de una sola línea de código, la cual retorna el valor actual de la variable privada, mas sin embargo un método set por su parte consta de por lo menos una línea de código que se modifica directamente del valor de la variable.

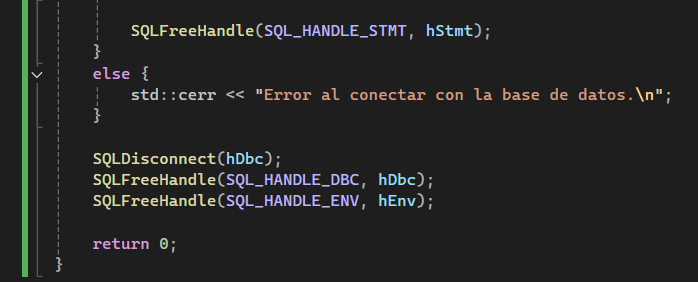
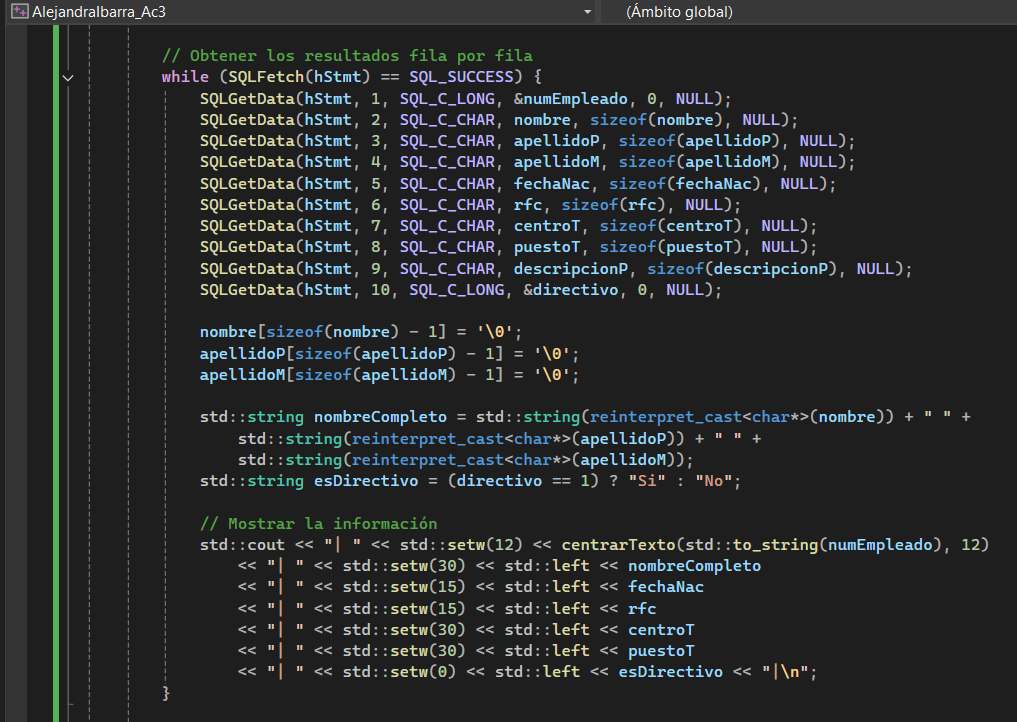
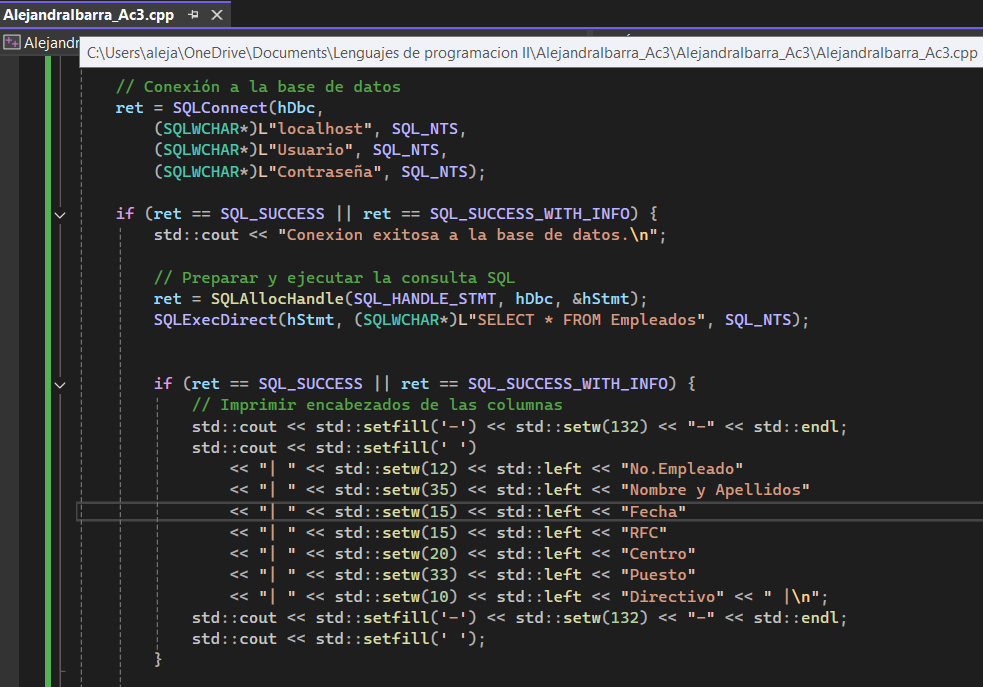
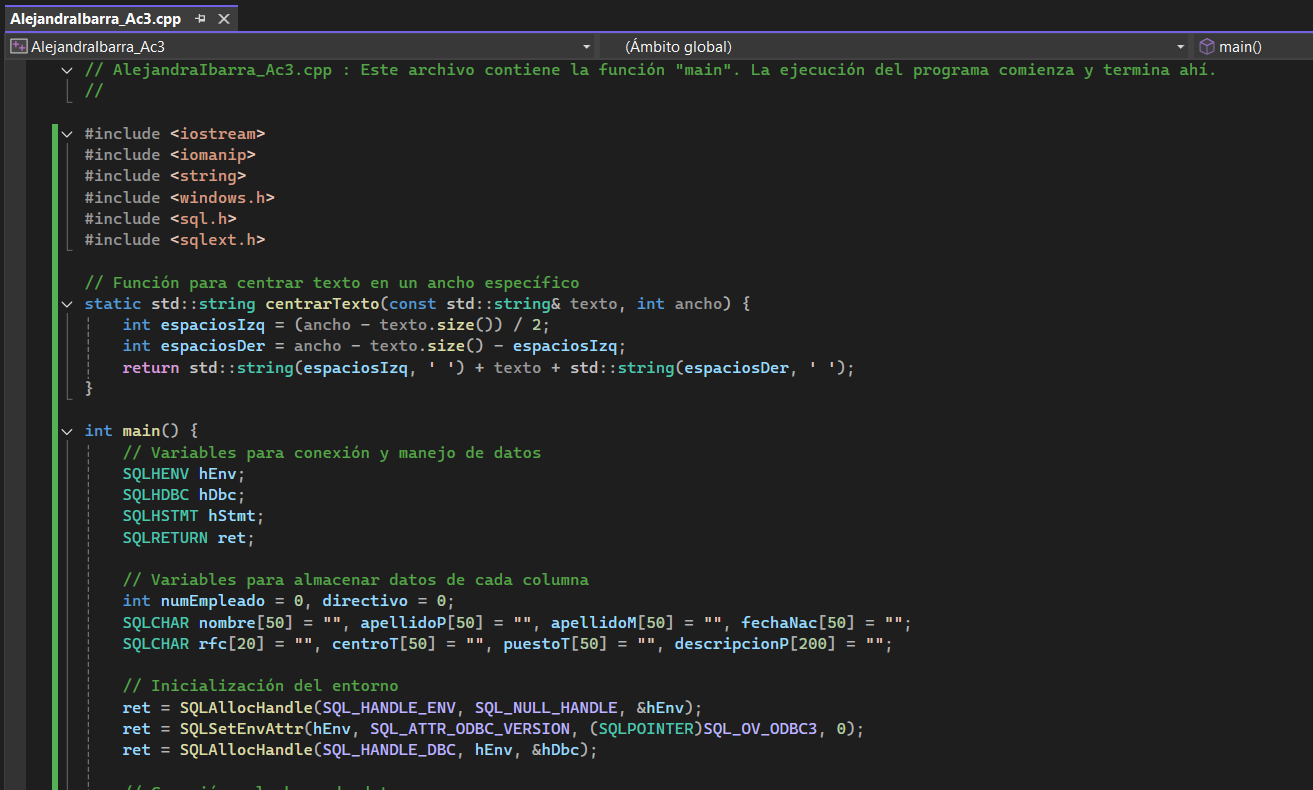
De tal manera, el método main puede utilizarlo uno de los métodos para enviar un valor de edad, el encapsulamiento permite tener control sobre los datos que manejamos con nuestros programas, para incrementar la seguridad de los datos se utilizan como modificadores de accesos mas sin embargo al quedar protegido por el modificador private y algunos de estos datos quedan aislados de procesos que en su momento puedan llegar a ocuparlos, por ello existen métodos conocidos como getters y setters que estos permiten gestionar el acceso a los datos de manera controlada y a partir de puntos específicos del programa.

Desarrollo

* Ejecución



* Codigo



Conclusion

Podemos llegar a la conclusión en esta actividad que el ejecutar el código de forma correcta nos puede ayudar a que nuestra tabla quede de una manera mas alineada al momento de jalar los datos de nuestra base.

Y de forma personal me pareció una actividad muy interesante solo que no me quedo de forma alineada la tabla, pero si con los datos correctos, el polimorfismo es una característica que no puede ser vista de forma aislada, por lo tanto, funcionan únicamente de forma conjunta como una parte de gran cuadro de relaciones.

La declaración de un método virtual puro ofrece solo la cabeza del método, normalmente no se ofrece una implementación de la función virtual pura en clase abstracta, pero es posible. Toda clase hija no abstracta continúa necesitando redefinir el método, pero la implementación ofrecida por la clase abstracta.

Referencias