Examen de fundamentos de programación

Seccion 1:

1. Todo lo que se halla en el repositorio remoto esta almacenado tanto en la nube de github como también en el almacenamiento interno del ordenador, contrario al local que si bien puede tener nuestro trabajo avanzado, no podría tener, en un caso teorico, todo lo demás que avancemos en otro ordenador, por lo que no importa que dispositivo uses, en el caso del remoto, mientras dichos documentos tengan la ultima versión almacenada en la nube, podras seguir lo que estabas realizando e incluso, guardar dicho progreso y seguir en tu ordenador local sin problemas.
2. El pull request generalmente es usado para verificar o/y comparar códigos en distintas ramas, ya fuesen ramas, o rama y main, y solicitar acciones cuales fuesen commits. Normalmente los conflictos de fusion suceden cuando un mismo documento proveniente de dos ramas se suben a la vez en una misma extensión o rama y coincidentemente se hallan en un mismo sector, renglón o línea, esto genera un merge y para resolverlo el usuario que presenta este merge tiene que separar lo señalado por el sistema, borrarlo o modificarlo.
3. Primeramente realizaría un Fetch para verificar que dichos cambios hallan sido subidos a la nube, si han sido subidos exitosamente, el sistema me pedirá hacer un pull para poder tener esos datos en la versión que tenga de dicho trabajo, realizando un pull, obtendré dicho documento y asi, tanto mi compañero como yo tendremos acceso a la nueva versión del trabajo en cuestión.

Seccion 2:

* Codigo:

int num1, extra, note;

string linea;

extra = 20 / 100;

bool esvalido;

Console.WriteLine("Ingrese la cantidad de respuestas correctas: "); linea = Console.ReadLine();

esvalido = int.TryParse(linea, out num1);

do

{

if (!esvalido)

{

Console.WriteLine("El dato ingresado no es valido, intente de nuevo");

return;

}

else

{

if (num1 >= 8)

{

note = num1 + extra;

Console.WriteLine("La nota obtenida califica como aprobatoria, felicidades.");

Console.WriteLine("Nota calificatoria: " + note);

Console.ReadKey();

}

else if (num1 == 7)

{

note = num1;

Console.WriteLine("Su nota califica como calificatoria:)");

Console.WriteLine("Nota calificada: " + note);

Console.ReadKey();

}

else

{

Console.WriteLine("Su nota no califica como aprobatoria");

Console.WriteLine("Nota calificada: " + num1);

Console.ReadKey();

}

}

} while (!esvalido);

* Diagrama:

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Seccion 3:

1. Codigo:

int producto, num1, num2, num3;

string linea;

Console.WriteLine("bienvenido, ingrese su pedido"); linea = Console.ReadLine();

producto = int.Parse(linea);

if (producto == 1)

{

Console.WriteLine("usted ha elegido comida, seleccione el producto que desea"); linea = Console.ReadLine();

producto = int.Parse(linea);

switch (producto)

{

case 1:

Console.WriteLine("usted ha elegido PAN \n precio: 3.50");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

case 2:

Console.WriteLine("usted ha elegido tallarines verdes \n precio: 5.00");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

case 3:

Console.WriteLine("usted ha elegido aji de gallina \n precio: 4.50");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

default:

Console.WriteLine("comando invalido");

break;

}

}

else if (producto == 2)

{

Console.WriteLine("usted ha elegido bebida, seleccione el producto que desea"); linea = Console.ReadLine();

producto = int.Parse(linea);

switch (producto)

{

case 1:

Console.WriteLine("usted ha elegido CHICHA \n precio: 2.50");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

case 2:

Console.WriteLine("usted ha elegido INCA COLA \n precio: 6.00");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

case 3:

Console.WriteLine("usted ha elegido PEPSI \n precio: 2.50");

Console.WriteLine("¿Desea continuar con su compra? s/n"); linea = Console.ReadLine();

if (linea == "s")

{

Console.WriteLine("compra realizada exitosamente, tenga buen dia");

Console.ReadKey();

}

else if (linea == "n")

{

Console.WriteLine("compra cancelada...");

Console.ReadKey();

}

else

{

Console.WriteLine("comando invalido-");

}

break;

default:

Console.WriteLine("comando invalido");

break;

}

}

else

{

Console.WriteLine("comando invalido, intente de nuevo");

}

* En este caso, si fuesen solo tres opciones podría basarme en generar un switch de tres casos y un default, de modo que en caso de que el cliente ingrese una opción no valida el sistema se lo señale tal que: “Opcion no valida, intente de nuevo” y se cierre la operacion. La otra opción podría ser trabajar con un Do-While de modo que al generar un booleano que verifique la validez del dato tengo un operador que me sirva para lo siguiente, dentro del Do-While desarrollar el código que ejecute el menú de elecciones disponibles basado en que si el usuario selecciona una opción entre 1-3 el sistema responda con la opción seleccionada mas precio y se le pregunte si desea continuar o no con cadenas de If/ elif, teniendo como base una cadena que señale la condición if (!esvalido) la cual detectara el error del usuario al tratar de digitar algo fuera de los parámetros.
* Resumiria con operadores el resultado total del precio tal que si producto1 + producto2 + producto3 = may >= $20

May – 10%

Claro que para que esto se vea bien ejecutado tendría que aplicarse en cada if de selección variando los operadores mencionados, siendo que, por ejemplo, si el primer if cuenta con el producto seria tal que asi:

Producto1 = ####

Si es el segundo if seria

Producto1 + producto 2 = total

Y asi sucesivamente, pudiendo variar incluso de orden pero no en efecto y resultado.