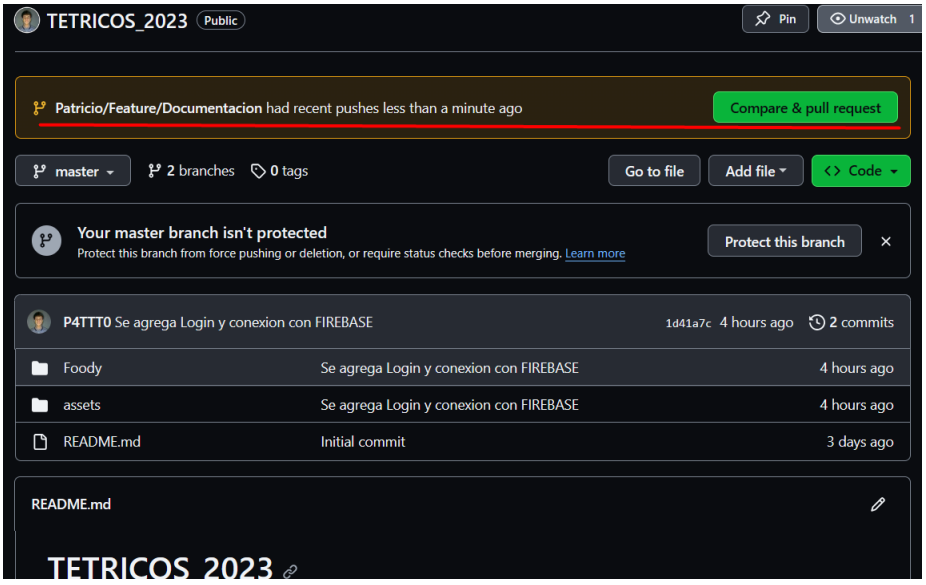
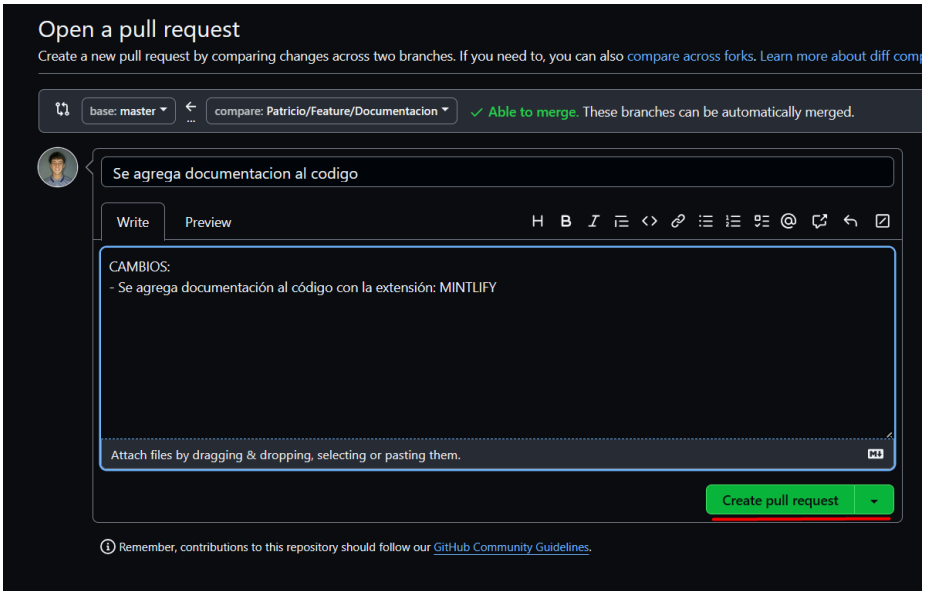
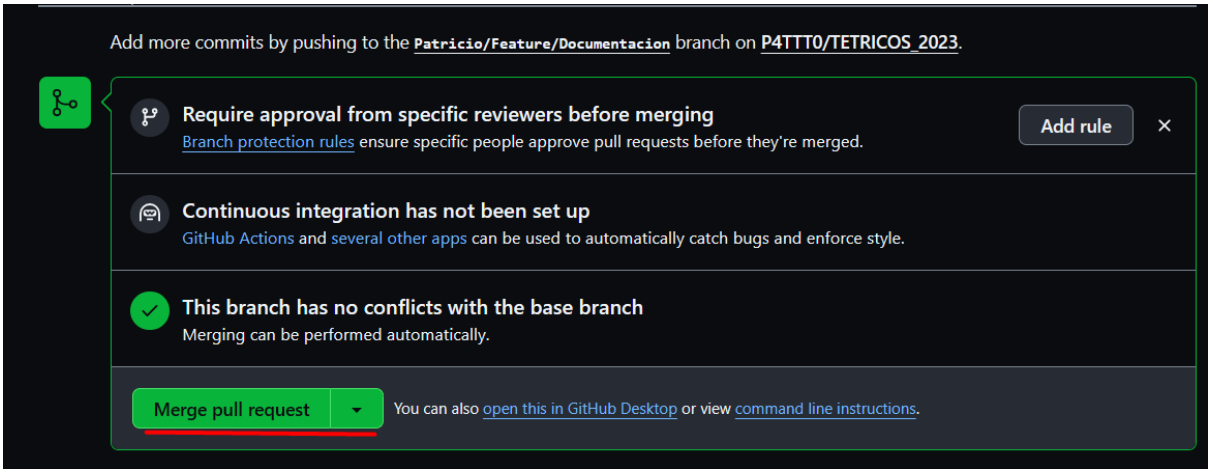
Trabajar en ramas:
[01] - Siempre verificar tener la ultima actualizacion de la rama MASTER:
[] - | git checkout master (Movernos a la rama master)
[] - | git pull origin master (Me traigo todos los cambios)
[02] - Creo una nueva rama (Nombre/Feature/Objetivo)
[] - | git checkout -b Patricio/Feature/Login
[03] - Realizo los cambios
[¡ATENTCION!] - Solo realizar UNICAMENTE cambios en los archivos donde los demas no esten trabajando, si dos o más personas trabajan sobre el mismo luego traeria problemas.
[04] - Envio los cambios
[] - | git add .
[] - | git commit -m "Descripcion del commit"
[] - | git push origin <nombreDeLaRama>
[05] - Realizar PR



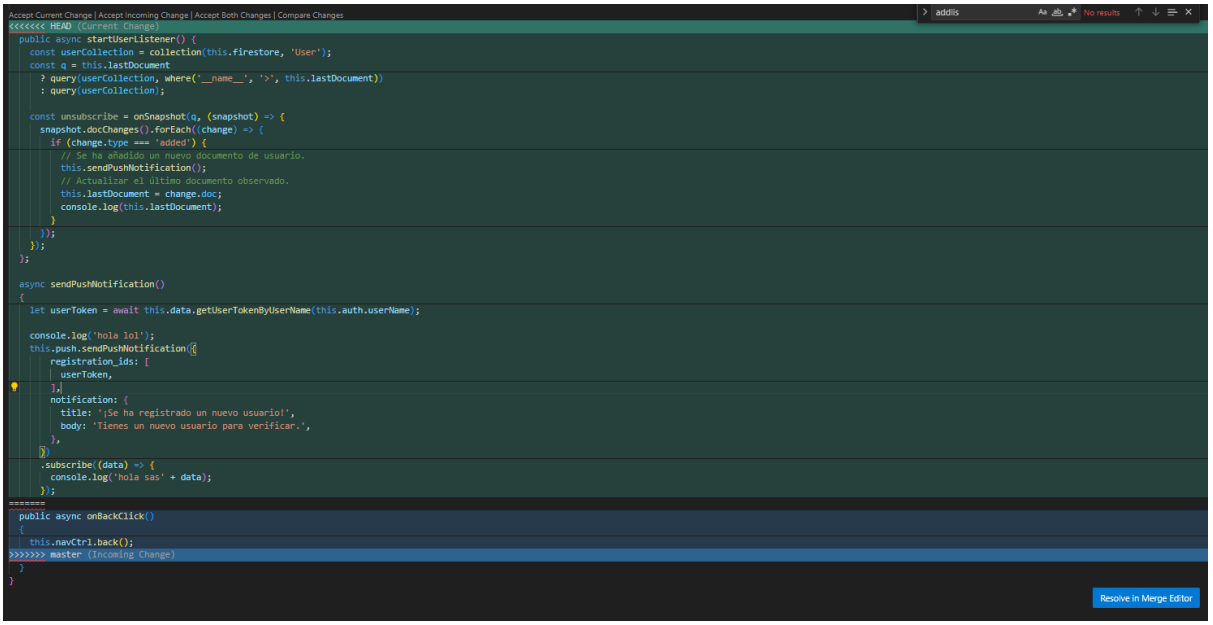


[06] - Esperar a que alguien revise el PR y realice el MERGE

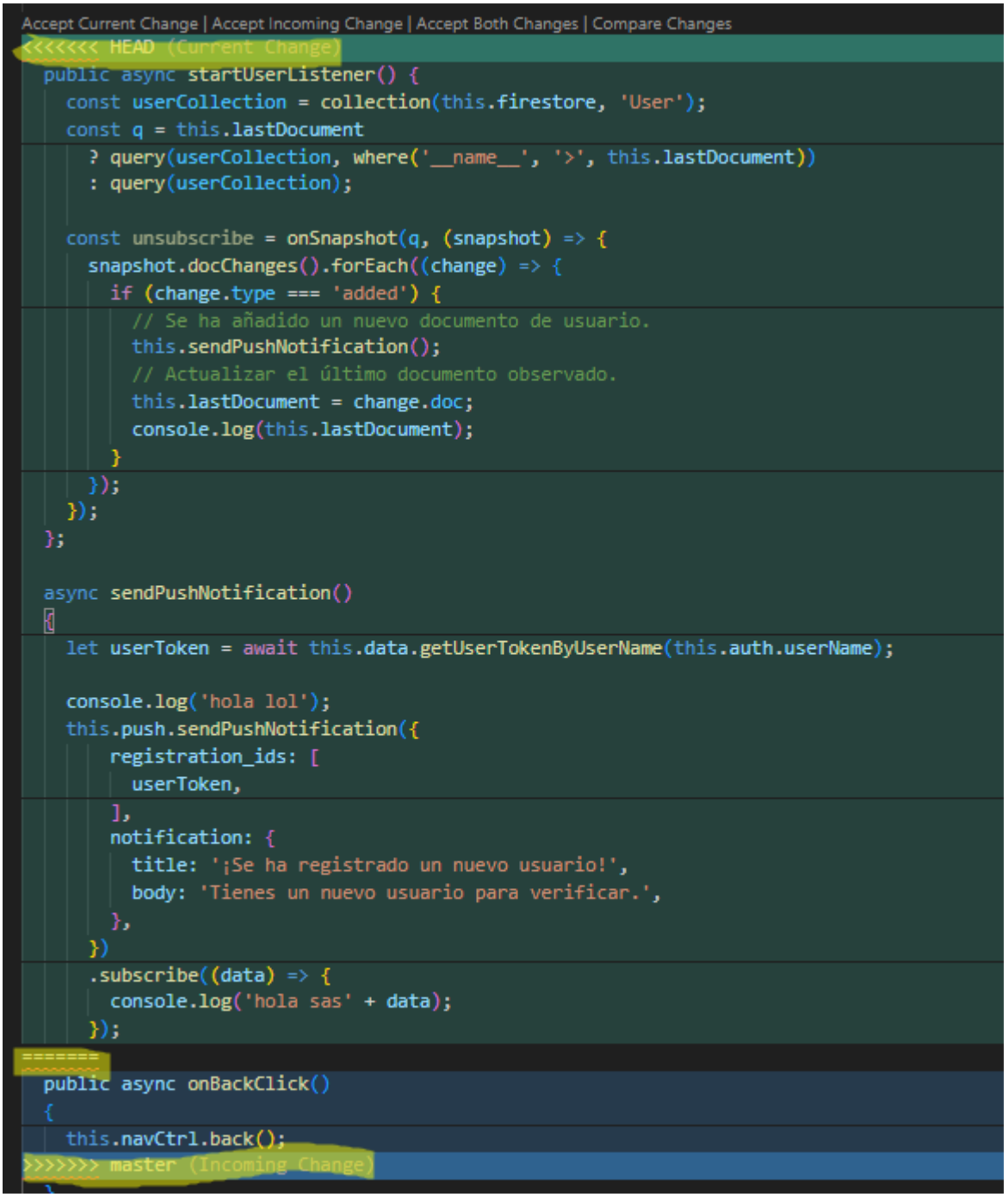


¿Qué debo hacer si mi rama se ha trabajado con la MASTER desactualizada?

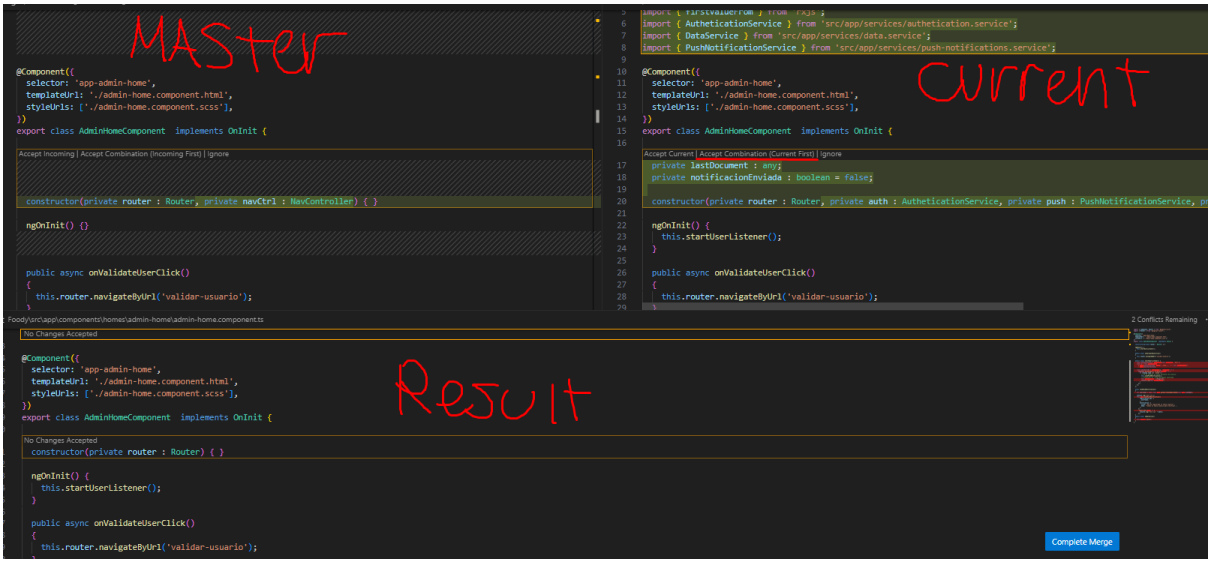
[01] - Vuelve a la rama master
`git checkout master`
[02] - Me actualizo la rama master
git pull origin master
[03] - Volves a tu rama
git checkout rama
[04] - Realizar el merge de tu rama con el master actualizado
git merge master
[!CUIDADO¡] - Lo más probable es que al realizar el merge ocurran CONFLICTOS, pero estos no son errores, solo hay que especificar cómo deben quedar los archivos.
Por ejemplo, en este archivo:



Podemos observar lo verde que son mis cambios hechos en mi rama y lo azul los cambios traídos de la master, simplemente hay que especificar con que nos queremos quedar.
Esto se puede hacer simplemente eliminando los textos agregados (Subrayados en amarillo):



También se puede realizar desde el MERGE EDITOR. Del lado izquierdo veremos el archivo de la master, del lado derecho veremos el archivo de nuestra rama y por debajo el resultado del MERGE. En este caso yo quiero quedarme con ambas modificaciones, la de la master y la de mi rama, de esta forma hago click en “Accept Combination” y luego completo el merge. Esto debemos repetirlo con cada archivo que tenga conflictos.


MASTER
Accept Incoming | Accept Combination (Incoming First) | Ignore
@Component({
selector: 'app-admin-home',
templateUrl: './admin-home.component.html',
styleUrls: ['./admin-home.component.scss'],
})
export class AdminHomeComponent implements OnInit {
constructor(private router : Router, private navCtrl : NavController) { }
ngOnInit() {}
public async onValidateUserClick()
{
this.router.navigateByUrl('validar-usuario');
}
import { AutheticationService } from 'src/app/services/authetication.service';
import { DataService } from 'src/app/services/data.service';
import { PushNotificationService } from 'src/app/services/push-notifications.service';
Current
Accept Current | Accept Combination (Current First) | Ignore
private lastDocument : any;
private notificacionEnviada : boolean = false;
constructor(private router : Router, private auth : AutheticationService, private push : PushNotificationService,
ngOnInit() {
this.startUserListener();
}
public async onValidateUserClick()
{
this.router.navigateByUrl('validar-usuario');
}

Foody\src\app\components\homes\admin-home\admin-home.component.ts
2 Conflicts Remaining
No Changes Accepted
Result
@Component({
selector: 'app-admin-home',
templateUrl: './admin-home.component.html',
styleUrls: ['./admin-home.component.scss'],
})
export class AdminHomeComponent implements OnInit {
No Changes Accepted
constructor(private router : Router) { }
ngOnInit() {
this.startUserListener();
}
public async onValidateUserClick()
{
this.router.navigateByUrl('validar-usuario');
Complete Merge