### PRACTICA Final

## Ramas permanentes

Los equipos suelen tener una o dos ramas permanentes, una donde está la versión estable de nuestro proyecto (suele ser la rama *master*), la que vamos a subir a nuestro servidor (o a GitHub Pages) y una intermedia donde ir fusionando los diferentes cambios/mejoras (nombres comunes para esta rama son *dev*, develop...). Esta segunda rama no siempre tiene código estable, pero **solo cuando lo tiene la fusionaremos hacia** master.

En el curso vamos a tener dos ramas permanentes: *master* y *dev*.

## Pull Request y revisión de código

Aunque hay diferentes formas de trabajo con ramas en el curso vamos a ver una muy común en las empresas: el equipo que se encarga de una tarea, crea la rama, ejecuta la tarea y la sube a la rama remota, pero no la fusiona: crea un **Pull Request**. De esta forma un equipo o algún miembro del equipo se encarga de revisar el código y dar el visto bueno a la fusión.

## Ramas puntuales

Cuando tenemos una tarea nueva (modificar o crear contenido, por ejemplo) crearemos una nueva rama solo para dicha tarea. Trabajaremos sobre ella con normalidad, haciendo los commits necesarios y subiéndola a su correspondiente rama remota.

## Code review

La revisión del código no es un asunto menor sino de los más importantes. Aquí no buscamos decir Ok y a correr, no. Aquí es donde tenemos la oportunidad de, con ojos frescos, revisar el código generado por nuestro/s compañeros para asegurar unos mínimos de compatibilidad, homogeneidad y calidad.

Cuando creamos un PR (o Pull Request) solemos asignar a alguien que será responsable de revisar nuestro código, un **reviewer**.

## Vamos a poner todo esto en contexto: Flujo de trabajo

Podríamos decir que a partir de este momento *dev* va a ser nuestra rama *máster* de desarrollo, y master será donde tendremos el código listo para entregar.

**EJERCICIO**

El trabajo lo vamos a realizar **por parejas**, en la que una de las 2 será el responsable de la aplicación, es decir va a hacer de jefe de proyecto, y será el que decida cuando reciba los PR de ver si están correctos y de subirlos a las ramas que el considere en cada momento.

Partimos de una base, html y css:

**index.html**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <link rel="stylesheet" href="main.css">

        <title>Face time!</title>

    </head>

    <body>

        <div class="page">

            <div class="head">

                <div class="face">:|</div>

            </div>

        </div>

    </body>

</html>

main.css

. .head {

    background: #fc0;

    height: 100px;

    position: relative;

    width: 100px;

  }

  .face {

    font-family: 'courier new', serif;

    font-size: 2.5em;

    font-weight: bold;

    left: 50%;

    position: absolute;

    top: 50%;

    transform: translate(-50%, -60%) rotate(90deg);

  }

Primero hay que crear el repositorio con estos dos archivos, dar permisos a nuestra compañera y crear las ramas *master* y *dev*.

A continuación, hay un listado de ejercicios/tareas que habrá que ir resolviendo. Usaremos lo que hemos visto de PR(Pull Request) pasando a *dev* cada tarea una vez completada. Cuando estén las 3 primeras en *dev*, haremos una subida a master, también por pull request.

Y por último cuando este la última en *dev* la subiremos a *master*, para dejar por fin nuestra aplicación terminada.

**Tarea #1: Centrar el careto**

Este señor está como apartado, en una esquina... eso no puede ser, debería estar centrado vertical y horizontalmente en la pantalla.

**Tarea #2: Cara redonda**

Un careto.. ¿es un careto si no es redondo? Hay que redondear esa cara para que sea un círculo perfecto.

**Tarea #3: Cara Sonriente**

¿No está un poco serio? Hagamos que sonría un poco.

**Tarea #4: Añadir funcionalidad para que guiñe**

Sería genial si al hacer click el careto guiñase un ojo. Hay que tener en cuenta que al dejar de hacer hover sobre el careto tiene que volver a la posición inicial.