

Configuración del Entorno de Trabajo

Daniel Elias Becerra A01208905
Hugo David Franco Ávila A01654856

Plataforma de comunicación: Slack

- Justificación

- Escogimos esta plataforma ya que permite generar varios canales o sub chats para los diferentes tasks o aspectos del proyecto a desarrollar.
- Mantiene un buen orden de los chats e información compartida.
- Tiene aplicaciones que se pueden integrar y conectar como Trello. Así se facilita el trabajo y todo se mantiene conectado.

Control de versiones: GitHub

Branching model

- Git Flow (Modelo de Driessen) [1]

- En donde master tiene el código de producción
- Y develop tiene el código de pre-producción
- Otras ramas con diferentes funcionalidades salen de develop y cuando se terminan se hace merge con develop
- También hay ramas hotfix que se usan cuando se quiera corregir rápidamente un error encontrado en el código que ya está en master
- También existe un rama llamada release que sale de develop y se utiliza para preparar el código antes de mandarlo a producción. Esta rama debe de hacer merge con master y develop cuando se hayan acabado de realizar cambios en la misma.

- Justificación

- Escogimos este modelo ya que permite que cada rama sea independiente la una de la otra y no haya errores de conflictos en el ciclo de vida de desarrollo del proyecto
- Además las ramas siguen una convención de nombres que permite que cada una se diferencie la una de la otra
- Permite también que cada miembro del equipo trabaje en su propio feature sin afectar el trabajo de los demás miembros y así evitar conflicts.
- También debemos de considerar que la integración continua se vuelve más difícil, por lo que como miembros del equipo debemos de asegurarnos de no ocasionar algún error en las ramas y la integración de nuestro trabajo.

Control de tareas: Trello

- Justificación

- Escogimos Trello ya que esta herramienta es fácil de entender y permite generar diferentes boards para cada aspecto a trabajar sobre el proyecto.

- Además se puede conectar con Slack, y a través de slack se pueden escribir comandos para generar tareas en los boards, asignar deadlines, encargados, entre otras cosas. Por lo que de esta manera la plataforma de control de tareas y de comunicación se mantienen conectadas y en sincronía.

Estándar de codificación

Seguiremos el estándar de JavaScript [2]

- **Nombres de variables**
 - camelCase
 - `var butterflyName = "monarca"`
 - `var name = "monarca"`
- **Espacio entre operadores**
 - `var result = 1 + 2 * 5;`
 - `var values = ["seed1", "seed2", "seed3"];`
- **Indentación y paréntesis**

```
function getLatitude() {
    return latitude;
}
```
- **Loops**

```
for (i = 0; i < 5; i++) {
    x += i;
}
```
- **Condicionales**

```
if (latitude < 7) {
    information = "in range";
} else {
    information = "outside range";
}
```
- **Objetos**

```
var person = {
    firstName: "Susana",
    lastName: "Pérez",
    age: 50,
};
```
- **Variables Globales y constantes**
 - UPPERCASE
 - LATITUDE

Otras consideraciones:

- Todo el código, como las variables y nombres de funciones serán en inglés.
- Los nombres de los archivos siempre deberán de incluir qué tipo de archivo de la estructura MVC son.

- userController.php
 - dataModel.php
 - homeView.html
 - homeStyle.css
- Además cada archivo deberá de ir en su carpeta correspondiente y cada de esas carpetas tendrá a qué módulo o grupo de requisito pertenece. Ejemplo
- ```
app
 views
 events
 editEventView.html
 addEventView.html
 calendarView.html
 controllers
 events
 insertEventController.php
 deleteEventController.php
```

## Referencias

- [1] <https://medium.com/@patrickporto/4-branching-workflows-for-git-30d0aaee7bf>
- [2] [https://www.w3schools.com/js/js\\_conventions.asp](https://www.w3schools.com/js/js_conventions.asp)