**Documento de Versionamiento: Carrera de Caballos**

**1. Introducción**

* **Objetivo del proyecto:** Desarrollar un programa en Java que permita visualizar una carrera de caballos y manipulación de esta.
* **Alcance:** Visualizar la carrera de caballos con un semáforo incluido, especificar el numero de competidores deseado, la interrupción de uno de los caballos por diferentes eventos y el conteo de ganadores.
* **Equipo de desarrollo:** Juan Sebastian Gracia Perez, Erika Juliana Puerto Corchuelo, Ana Karina Roa Mora.

**2. Configuración del Repositorio**

* **Plataforma:** GitHub.
* **URL del repositorio:** https://github.com/ErikaPuerto/Carrera
* **Estructura de carpetas:**
  + src: Código fuente de la aplicación.
  + docs: Integrantes
  + specs: Diagrama de clases UML, documento explicativo de versionamiento.

**3. Flujo de Trabajo**

* **Rama principal (master):** Contiene la primera versión en donde se ve gran avance de caballos, permitiendo visualizar una beta del funcionamiento, con texto primitivo en vez de imágenes funcionales.
* **Ramas de desarrollo:**
  + rama1(Juliana): Está rama contiene las primeras versiones de la vista como la posibilidad de ver las imágenes de los caballos, así como la modificación de los botones principales como “interrumpir” e “iniciar carrera”.
  + rama2(Juan): En esta rama se corrige la organización del código en función al MVC y los principios SOLID para evitar acoplamiento y para que sea un código mejor estructurado, así mismo se implementan los “rivales”.
  + rama3(Karina): En esta rama se encuentran los cambios principales de la vista, así como las correcciones pertinentes de la misma.
* **Proceso de desarrollo:**
  + Cada desarrollador crea una rama a partir de la rama en la cual va a trabajar y a cada uno se le da una tarea principal para realizar y si uno logra su parte se la va pasado a los otros para que tengan códigos más completos y visualizar que funcione bien.
  + Se realizan commits a cambios o mejoras que se creen importantes, como los requisitos la optimización de los métodos, una reorganización mejorada, una nueva implementación en el programa, corrección de posibles complicaciones, etc.
  + Se realizan revisiones de cada rama para verificar que todo estuviera funcionando correcto y cumpliendo con el modelo de tres capas para tener unidas las cuatro ramas completas fusionarlas y subirla a la rama master.

**4. Convenciones de Commit**

* **Formato:** Se da una breve explicación directa y detallada de los cambios realizados.
  + Ejemplo: Modificación Caballos a Imagen

5. Procedimiento de carga de commits a GitHub:

* Si aún no se tiene clonado el repositorio de git, se usa:

git clone https://github.com/ErikaPuerto/Carrera.git

* Verificación de la rama actual: Nos aseguramos de estar en la rama correcta usando:

git branch

* Se realizan los cambios: Modificamos los archivos necesarios en el proyecto.
* Se agregan los cambios: Para agregar los cambios realizados:

git add .

* Si es necesario crear una nueva rama antes de hacer el commit, puedes hacerlo con

git checkout -b nueva-rama

* Hacer el commit: Una vez que los cambios están listos, se crea un commit con un mensaje descriptivo:

git commit -m "Descripción del cambio”

* Empuja los cambios a la rama en GitHub: Finalmente, empuja los cambios a la rama específica en GitHub:

git push origin nombre-de-la-rama

* Se revisa en GitHub si se subió correctamente el commit.

**6. Historial de Versiones**

* **Gráfico de ramas manejadas:**
* **Tabla de Commits importantes:**

|  |  |  |
| --- | --- | --- |
| **Commit** | **Fecha** | Descripción |
| Primera Version | 29/Oct/2024 | Se realizó la primera versión del programa con las clases principales e funcionamiento principal |
| Modificación Interrumpir e Iniciar Carrera | 29/Oct/2024 | Se modificaron los botones interrumpir e iniciar carrera para su correcto funcionamiento |
| Correccion de MVC y Mejora de parte Gráfica | 30/Oct/2024 | Se reorganizó el código para que se adapte al modelo MVC y SOLID de mejor manera |
| Modificacion Caballos a Imagen | 31/Oct/2024 | Se permitió la entrada de imágenes, entre ellas la de los caballos |
| Vista Actualizada | 01/Oct/2024 | Se evidencia la implementación de la interfaz para mejor calidad estética e interacción con el usuario. |
| Agregar Que Haya Rivales | 03/Nov/2024 | Se modificó el código para que reconozca a los demás caballos como rivales y se detengan cuando uno de ellos llegue a la meta |
| Versión Final | 04/Nov/2024 | Contiene la documentación del programa, archivos necesarios fuera del código del proyecto (diagrama UML, versionamiento e integrantes) y mejora en botón Interrumpir |

* **Hitos importantes:**
  + Primera versión con funcionalidad primitiva de la carrera
  + Separación de responsabilidades
  + Implementación de lógica faltante
  + Implementación de la interfaz gráfica.
  + Revisión del funcionamiento y requerimientos.
  + Corrección de bugs críticos.

**7. Herramientas Utilizadas**

* **Git:** Control de versiones.
* **GitHub:** Plataforma de hospedaje de repositorios.
* **NetBeans:** Entorno de desarrollo integrado.
* **Java:** Lenguaje de programación.
* **Java con Maven:** Maven para la utilización de las pruebas unitarias de las clases controladoras.

**8. Conclusiones**

En este proyecto, se lograron avances significativos en la estructura del código y las funcionalidades, como lo es el uso de hilos de ejecución para ejecutar tareas en paralelo, optimizar operaciones, mejorar la experiencia del usuario y facilitar procesamiento de datos. De igual manera, se estableció una base sólida para la lógica del programa y se adoptó la arquitectura MVC y los Principios SOLID, lo que facilitará el mantenimiento futuro. Por último, el uso de Git fue clave para gestionar el desarrollo de manera eficiente.