

**Visualización de Resultados**

**Índice:**

Introducción………………………………………...............................................3

Resumen………………………………………………..........................................4

Descripción del sistema………………………………………............................5

Entorno de desarrollo………………………………………………………………6

Gestión del código fuente………………………………………………………..7

Gestión de la construcción e integración continua ……………………......8

Gestión de cambio, incidencias y depuración…………………………........9

Gestión de liberación, despliegue y entregas…………………………........10

Mapa de herramientas …………………………………………………………..11

Conclusiones y trabajo futuro……………………………………………….......12

Criterios de aceptación del proyecto………………………………………....13

Introducción:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre** | **Apellidos** | **Grupo** | **Curso** | **Nombre del proyecto** | **ID opera** |
| Belén | Ramos Gutiérrez | 2 - Tarde | 4 | Visualización de resultados | 777 |
| Diego | Rodríguez Fernández | 2 - Tarde | 4 | Visualización de resultados | 779 |
| David | Soto Sánchez | 2 - Tarde | 4 | Visualización de resultados | 780 |
| Adrián | Luque Montoya | 2 - Tarde | 4 | Visualización de resultados | 781 |
| Emilio | Beltrán Martínez | 2 - Tarde | 4 | Visualización de resultados | 782 |

|  |
| --- |
| **Enlace al repositorio del proyecto, donde encontraras todas las tareas, documentos y tareas.** |
| <https://github.com/belramgut/EGC-VisualizacionDeresultados> |

Resumen:

*Se sintetizará de un vistazo lo hecho en el trabajo y los datos fundamentales. Se usarán palabras para resumir el proyecto presentado. Un buen resumen contendrá el problema que se ha tratado, por qué es un problema importante, la solución que se ha planteado, por qué es una buena solución y algunas conclusiones del trabajo.*

Introducción y contexto:

*Se pondrá en contexto el proyecto elegido y los aspectos fundamentales para entender el resto del documento. La introducción es como un resumen ampliado.*

Descripción del sistema:

*Se explicará el sistema desarrollado desde un punto de vista funcional. Se hará una descripción tanto descriptiva como técnica de sus componentes y su relación con el resto de subsistemas. Habrá una sección que enumere explícitamente cuáles son los cambios que se han desarrollado para el proyecto. En otra sección se presentará la planificación del trabajo y qué tareas se han hecho. Haciendo hincapié en el reparto de tareas. En este sentido que esté sentido es importante que estén explícitas las iteraciones funcionales que se van a realizar por cada parte del miembro del equipo. Tenga muy en cuenta lo expresado en la sección 2.2.2 en cuánto a división de tareas.*

*Sección 2.2.2:*

*Todos los componentes del equipo deben ser desarrolladores en el proyecto. No se*

*puede dividir el equipo para que unos hagan desarrollo y otros hagan documentación*

*por ejemplo. Para poder poner en práctica los conocimientos debe todo el mundo ser*

*desarrollador/a y trabajar en paralelo. Se podrá cargar más en una persona u otra la*

*parte de desarrollo, pero en ningún caso se podrá quitar a alguien de esta labor.*

Entorno de desarrollo:

*Debe explicar cuál es el entorno de desarrollo que ha usado, cuáles son las versiones usadas y qué pasos hay que seguir para instalar tanto su sistema como los subsistemas relacionados para hacer funcionar el sistema al completo.*

Gestión del código fuente:

*Se explicarán los procesos, técnicas y herramientas para la gestión del código del* *proyecto. Evite poner información de las herramientas en sí que se pueda encontrar en fuentes bibliográficas o internet. Si es del caso haga referencia a ellas. Céntrese en los aspectos particulares de su proyecto, por ejemplo, ¿cómo se gestionan las ramas en el código? ¿cómo se aplica un parche (patch)? ¿cómo se aprueban los cambios? ¿qué roles existen en la gestión del código?, ¿qué políticas de nombre y estilo se utilizan en el código fuente? ¿cuál es el usage model del repositorio de código? Cada uno de estos apartados deben estar divididos en secciones.*

Gestión de la construcción e integración continua:

*Gestión de la construcción e integración continua: similar al apartado anterior se*

*definirán los procesos que se usan a la hora de construir el proyecto, ¿qué herramientas se usan? ¿cómo se usan? ¿cada cuánto tiempo se realiza una construcción del proyecto? ¿Qué mecanismos IC se usan? ¿cómo?, etcétera.*

Gestión de cambio, incidencias y depuración:

*¿Qué mecanismos de depuración se usan? ¿cómo se gestionan los cambios? ¿qué procesos? ¿qué roles? ¿qué estados se manejan? ¿qué políticas para descartar, fomentar o retardar un cambio? La gestión de incidencias debería contener explícitamente dos partes. Una de cómo se han gestionado las incidencias internas y otro el cómo se han gestionado y se ofrece protocolo para gestionar las incidencias externas tanto las recibidas como las que se reporten a otros subsistemas. Cuando una incidencia esté relacionada con un commit, señalar el commit dentro de la propia incidencia*

Mirar apartado 2.4 del trabajo de EGC

Gestión de liberaciones, despliegue y entregas:

*¿Qué elementos del proyecto son “entregables”? ¿cómo se generan? ¿cómo se identifican? ¿cómo se gestiona la publicación, la liberación y entrega? ¿dónde se entrega? ¿qué roles existen en la entrega? ¿qué mecanismos de despliegue se definen? ¿qué procesos? ¿qué plataformas? ¿qué herramientas?*

Mapa de herramientas:

*Debe dar un esquema de cómo se conectan las herramientas que se usan en el proyecto, qué relaciones tienen o qué relaciones propondría añadir. No olvide explicar bien el mapa de herramientas. Se trata del mapa de herramientas de gestión de la configuración. El mapa de herramientas que se usen para el desarrollo (como bases de datos u otros) no es necesario que aparezca aquí*

Conclusiones y trabajo futuro:

*Enunciar algunas conclusiones. Añadir una parte sobre las mejoras que se proponen para el futuro (curso siguiente) y que no han sido desarrolladas en el sistema actual.*