**Identificación**

****

|  |  |
| --- | --- |
| **Versión** | 1.0 |
| **Proyecto** | Giroplas |
| **Fecha** | 16/05/2015 |
| **Elaborado por** | Luis Eduardo Correa Romero |
| **Localización del Documento** | <https://github.com/lalovicompu/giroplas> |
| **Documento base** |  |
|  |  |
| **Autorizaciones** | |
| **Nombre** | **Fecha Autorización** |
| LECR | 18/05/2015 |
|  |  |
|  |  |
|  |  |

**Control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Control de Cambios** | | | |
| **Versión** | **Fecha** | **Participante** | **Descripción del Cambio** |
| 1.0 | 21/05/2015 | LECR | Ciclo de vida |
|  |  |  |  |
|  |  |  |  |

**Ciclo de vida**

La importancia de saber por qué utilizar este estilo de ciclo de vida ya que el seguimiento de las fases son las más prácticas para poderlas implementar en nuestro proyecto.

**Análisis:** con base a los requerimientos que se estuvieron desarrollando poderlos ir enfocando en cada una de las demás fases.

**Diseño:** en esta sección se estarán desarrollando lo que es el logotipo de la empresa, toda la estructura del sitio.

**Codificación:** aquí se estará programando todo lo que es el sitio y nuestra base de datos MYSQL.

**Pruebas:** se implementaran las pruebas necesarias para que se puedan diagnosticar rápidamente los problemas que se obtengan y poder solucionarlos rápidamente.

**Cierre del Proyecto:** aquí se concluye exitosamente en el funcionamiento del sitio.

**Justificar el patrón o patrones de diseño a utilizar**

En esta parte se utilizara el patrón de diseño el Singleton está diseñado para restringir la instanciación de una clase o valor de un tipo a un solo o eventualmente muy pocos objetos. Resulta muy útil cuando por ejemplo se necesita un único objeto que coordina acciones en un sistema. Algunas veces se utiliza también en aplicaciones en donde resulta más eficiente trabajar con uno o muy pocos objetos es por ello que se utilizara este patrón de diseño porque es sencillo de manejar y se puede adecuar a nuestro sitio.