Documento integrador

Proyecto Final de Construcción de Software

Equipo #

Proyecto: Software de Ordenamiento y Búsquedas.

Integrantes:

* Angel Gonzales
* Pedro Cauich
* Alejandro Cauich
* Josue Torres

Lista de practicas de construccion software:

* Establecer las bases de éxito de la construcción
* Estandares de Codificacion
* Técnicas de Construcción de variables
* Aspectos generales del uso de variables
* Nombramiento de variables
* Convenciones de nomenclatura
* Uso de tipos de datos fundamentales (Lineamientos)
* Tecnicas de Organizacion de sentencias
* Uso adecuado de condicionales
* Lineamientos de bucles de control
* Tecnicas de construccion de procedimientos ( rutinas )
* Cohesión
* Construccion de clases
* Abstract Data Types (ADTs)
* Herencia
* Técnicas Documentación de código
* Refactorización.
* Movimiento de funciones entre objetos

\* = señalamiento de lo que se ha realizado o lo que no se ha aplicado, cabe destacar que esto puede (y debe) cambiar con futuras versiones

| UNIDAD 1: Fundamentos y Técnicas de Construcción de software | | | |
| --- | --- | --- | --- |
| Técnicas de construcción de variables y tipos de datos fundamentales | | | |
| Prácticas de construcción | Se realizó | No aplica | |
| Estandares de Codificacion | \* |  | |
| Ámbito de las Variables |  |  | |
| Persistencia |  | \* | |
| Binding Time | \* |  | |
| Relación entre tipos de datos y estructuras de control | \* |  | |
| Nombramiento adecuado de variables ( Java ) | \* |  | |
| Convenciones de nomenclatura | \* |  | |
| Lineamiento de números en General | \* |  | |
| Lineamientos de caracteres y cadenas |  | \* | |
| Lineamientos de variables booleanas | \* |  | |
| Lineamientos de tipos enumerados |  | \* | |
| Lineamientos de constantes de nombre |  | \* | |
| Lineamientos de Arreglos | \* |  | |
| Tecnicas de organizacion de sentencias | | | |
| Lineamientos de orden específico |  | \* | |
| Lineamientos sin orden específico | \* |  | |
| Tecnicas de construccion de estructuras de control de flujo | | | |
| Uso de condicionales if-then | \* |  | |
| Condicional if-then-else | \* |  | |
| case |  | \* | |
| Bucles de control | \* (For) |  | |
| Técnicas de construcción de procedimientos. | | | |
| Rutinas de alta calidad |  | \* | |
| Tipos de Cohesión ( Diseño a nivel rutina) |  | \* | |
| Lineamientos para buenos nombres de rutinas |  | \* | |
| Parámetros |  | \* | |
| Tecnicas de construccion de clases | | | |
| Abstract Data Types (ADT) | \* |  | |
| lineamientos para Interfaces de clases | \* |  | |
| Contención |  | \* | |
| Herencia |  | \* | |
| Lineamientos para miembros datos y funciones |  | \* | |
| Constructores |  | \* | |
| Técnicas de documentación de código | | | |
| Comentarios |  | \* | |

| UNIDAD 2: Refactorización de Software | | | |
| --- | --- | --- | --- |
| Composición de métodos | | | |
| Prácticas de construcción | Se realizó | No aplica | |
| Extract Method | \* |  | |
| Inline Method |  | \* | |
| Inline Temp |  | \* | |
| Replace Temp with Query | \* |  | |
| Introduce Explaining Variable |  | \* | |
| Split Temporary Variable |  | \* | |
| Remove Assignments to Parameters | \* |  | |
| Replace Method with Method Object |  | \* | |
| Substitute Algorithm | \* |  | |
| Movimiento de funciones entre objetos | | | |
| Move Method | \* |  | |
| Move Field |  | \* | |
| Extract Class |  | \* | |
| Inline Class |  | \* | |
| Hide Delegate | \* |  | |
| Remove Middle Man |  | \* | |
| Introduce Foreign Method | \* |  | |
| Introduce Local Extension |  | \* | |
| Organización de datos | | | |
| Self Encapsulate Field | \* |  | |
| Replace Data Value with Object |  | \* | |
| Change Value to Reference |  | \* | |
| Change Reference to Value |  | \* | |
| Replace Array with Object |  | \* | |
| Duplicate Observed Data |  | \* | |
| Change Unidirectional Association to Bidirectional |  | \* | |
| Change Bidirectional Association to Unidirectional |  | \* | |
| Replace Magic number with Symbolic Constant | \* |  | |
| Encapsulate Field | \* |  | |
| Encapsulate Collection |  | \* | |
| Replace Record with Data Class |  | \* | |
| Replace Subclass with Fields |  | \* | |
| Replace Type Code with Class |  | \* | |
| Replace Type Code with Subclasses |  | \* | |
| Replace Type Code with State/Strategy |  | \* | |
| Simplificación de expresiones condicionales | | | |
| Decompose Conditional |  | \* | |
| Consolidate Conditional Expression | \* |  | |
| Consolidate Duplicate Conditional Fragments | \* |  | |
| Remove Control Flag |  | \* | |
| Replace Nested Conditional with Guard Classes |  | \* | |
| Replace Conditional with Polymorphism |  | \* | |
| Introduce Null Object |  | \* | |
| Simplificación de las llamadas a métodos (llamadas simples a métodos) | | | |
| Rename Method | \* |  | |
| Add Parameter |  | \* | |
| Remove Parameter |  | \* | |
| Separate Query from Modifier |  | \* | |
| Parameterize Method | \* |  | |
| Replace Parameter with Explicit Methods |  | \* | |
| Preserve whole Object |  | \* | |
| Replace Parameter with Method |  | \* | |
| Introduce Parameter Object |  | \* | |
| Remove Setting Parameter |  | \* | |
| Remove Setting Method |  | \* | |
| Hide Method |  | \* | |
| Replace Constructor with Factory Method |  | \* | |
| Encapsulate Downcast |  | \* | |
| Replace Error code with Exception |  | \* | |
| Replace Exception with Test | \* |  | |
| Manejo de la generalización | | | |
| Pull up Field |  | \* | |
| Pull Up Method |  | \* | |
| Pull Up Constructor Body |  | \* | |
| Push Down Method |  | \* | |
| Push Down Field |  | \* | |
| Extract Subclass |  | \* | |
| Extract Superclass |  | \* | |
| Extract Interface |  | \* | |
| Collapse Hierarchy |  | \* | |
| Form Template Method |  | \* | |
| Replace Inheritance with Delegation |  | \* | |
| Replace Delegation with Inheritance |  | \* | |