

Documento de buenas prácticas de desarrollo

Propósito

Establecer un marco universal para la implementación de un desarrollo y mantenimiento de software que facilite el mantenimiento, escalabilidad y la incorporación de recursos con un tiempo mínimo de implementación. Como también corregir malos hábitos en el proceso de diseño de esté mismo.

Nomenclatura

Se describen los criterios generales relacionados con la forma de nombrar las distintas entidades usando los lenguajes de programación php, html, css, etc.

Estilos de capitalización

Hay una serie de posibles criterios que se utilizarán para facilitar a la distinción entre variables, clases, constantes, métodos, etc. Las opciones por las cuales optamos son:

- Tipo “Pascal”: La primera letra de cada palabra que compone el identificador se pone en mayúscula. Es el tipo que se utiliza en la mayor parte de los casos (nombre de clases, de métodos, de espacios de nombres, facturación, reservas, etc.). Por ejemplo: FechReserva.
- Tipo “camel”:
- Igual que el anterior pero la primera es en minúscula. Se utiliza para nombres de variables locales y parámetros de métodos. Por ejemplo: valorTotal.
- Mayúscula: todo el identificador se pone en mayúscula. Solo se usa para constantes o macros definidas en la documentación: Por ejemplo: DEBUG.

Nombre de entidad

Para la elección de los nombres de las entidades que forman parte del desarrollo (ficheros source, librerías, procedimientos, variables, parámetros, proyectos, assemblies, etc. Se seguirán las siguientes reglas generalmente:

- Los nombres deben ser palabras mnemotécnicas o combinación de ambas relacionados con los nombres de las entidades del mundo real y con su función dentro del modelo lógico de la aplicación. No harán referencia a la forma de implementarse.
- Se formarán por la concatenación de una o más palabras sin usar separadores, espacios ni caracteres especiales.
- Pueden contener letras de la A a la Z, dígitos del 0 al 9. El primero carácter ha de ser una letra, aunque el carácter “_” podrá usarse como primer carácter, pero solo para variables privadas que corresponder con una propiedad pública con igual nombre.
- El uso de mayúsculas y minúsculas debe seguir unas normas que se comentara en cada caso.
- Elija palabras que no creen ambigüedades. Tenga en cuenta que una palabra puede tener varios significados y un significado puede expresarse con varias palabras.
- Elija identificadores intentando ser claro, no ameno ni gracioso.
- Nunca piense de un identificador que "por el contexto se deduce para que sirve"; elija identificadores suficientemente indicativos incluso fuera de contexto.

- No hay una longitud máxima ni mínima para los identificadores, aunque el sentido común pondrá unos límites (no más de 263 palabras).
- Utilice el mismo idioma para todos los identificadores.
- Podrán hacerse excepciones a estas reglas con un motivo bien y que deberán documentarse adecuadamente.

NameSpace

- Los nombres se elegirán según las normas generales.
- Se usará capitalización tipo Pascal.
- El nombre del espacio de nombres se elegirá conforme a la especificación del documento de Arquitectura.

Clases & Interfaces

- Los nombres se elegirán según las normas generales.
- Los nombres deberán ser palabras o frases formadas por nombres y/o sintagmas nominales (se evitarán verbos, artículos, etc.).
- Se usará capitalización tipo Pascal.
- No llevarán prefijo de tipo, pero las clases que sean interface, por convenio, irán precedidas de una "I" mayúscula.
- Cuando la clase derive de otra, ménciñela en el nombre (siempre y cuando eso tenga sentido). Por ejemplo, la clase derivada de "Usuario" podría ser "UsuarioAutenticado".
- Ciertos tipos de clases tienen sufijos universalmente aceptados:
 - Exception para excepciones. Ejemplo FatalException
 - Collection para colecciones. Ejemplo ActorCollection
 - Delegate para delegados.

- Attribute para atributos.
- Base. Clases abstractas que implementan cierta funcionalidad común a varias clases concretas.

Propiedades de Clases

- Los nombres se elegirán según las normas generales.
- Los nombres deberán ser palabras o frases formadas por nombres y/o sintagmas nominales (se evitarán verbos, artículos, etc.).
- Se usará capitalización tipo Pascal.
- Si se trata de una propiedad pública de la clase que encapsula a una variable privada de la clase, ambas deberán llamarse igual, anteponiendo el prefijo “_” a la variable privada.

Métodos

- Los nombres se elegirán según las normas generales.
- Habitualmente serán verbos, sintagmas verbales o combinaciones de ellos con nombres o sintagmas nominales.
- Los nombres para miembros públicos pueden depender del contexto, es decir de la clase a la que pertenecen; no así los que tengan otra visibilidad. Por ejemplo: un método publico Borrar de la clase Usuario podría llamarse "EliminarReserva" por claridad, mientras que uno privado bastaría con llamarlo "Borrar".
- Se usará capitalización tipo Pascal.

Variables

- Los nombres se elegirán según las normas generales.
- Los nombres deberán ser palabras o frases formadas por nombres y/o sintagmas nominales (se evitarán verbos, artículos, etc.).
- Se usará capitalización tipo Camel.
- Los identificadores de matrices y demás colecciones deberán estar en plural.

Constantes

- Los nombres se elegirán según las normas generales.
- Los nombres deberán ser palabras o frases formadas por nombres y/o sintagmas nominales (se evitarán verbos, artículos, etc.).
- Se usará el mismo criterio de capitalización que para las variables: Pascal para las constantes públicas y Camel para las privadas. También puede usarse solo mayúsculas en cuyo caso se podrá utilizar el carácter "_" para separar palabras.