Construyendo DSLs en Java

Aprovechando las nuevas herramientas del lenguaje

Antonio Muñoz

¿Quien soy?

- Programo en Java desde Java 1.1.
- Actualmente trabajo en https://clarity.ai como backender.
- Me encontraréis en:
  - Mastodon: https://jvm.social/@tonivade
  - Github: https://github.com/tonivade
  - Blog: https://blog.tonivade.es
  - Email: me@tonivade.es

commit

JMAD

17 de Mayo de 2025.

JMad OpenSpace

En conmemoración del 30 aniversario de Java, MadridJUG quiere rendir homenaje a este lenguaje icónico organizando un evento especial para su comunidad. Para ello, llevará a cabo un Open Space, un formato dinámico y participativo que permitirá a los asistentes compartir sus experiencias, debatir sobre las tecnologías del ecosistema de la JVM y explorar juntos los avances más recientes en el mundo del desarrollo en Java.

https://jmad.madridjug.es/tickets

commit

me@tonivade.es

Empecemos por el principio

- Qué es un DSL?
- Domain Specific Language.
- Está especializado para un dominio específico.

commit

me@tonivade.es

Ventajas

- Vocabulario reducido.
- Sintaxis concisa.
- Mayor expresividad.
- Mayor productividad.

commit

me@tonivade.es

Tipos de DSLs

- Externos.
- Internos o embebidos.

me@tonivade.es

commit

DSLs externos

- Están implementados con un interprete/compilador independiente.
- Ejemplos: Gherkin, SQL, HTML.
- Más complejos de implementar.
- Pero más potentes.

commit

me@tonivade.es

Gherkin

```
Feature:  Login functionality of social networking site Mastodon.
Given:  I am a mastodon user.
When: I enter username as username.
And I enter the password as the password
Then I should be redirected to the home page of mastodon
```

commit

me@tonivade.es

SQL

```
select
  people.name,
  people.age
from people
where people.age > 18
order by people.age ASC
```

commit

me@tonivade.es

HTML

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

commit

me@tonivade.es

DSLs internos

- Están implementados como parte del propio lenguaje donde se van a usar.
- Más sencillos de implementar.
- Están limitados a las capacidades del lenguaje anfitrión.

commit

me@tonivade.es

DSLs internos (II)

- Hay dos tipos de DSLs internos:
  - Estáticos
    - Ejemplos: JOOQ, Assertj.
  - Dinámicos
    - Ejemplos: Gradle.

me@tonivade.es

commit

De qué voy a hablar hoy?

- Un breve resumen de las novedades más relevantes en Java:
  - Records.
  - Pattern matching.
  - Sealed interfaces.
- DSLs internos estáticos implementados en Java. Muy brevemente.
- DSLs internos dinámicos implementados en Java. Más extensamente.

commit

me@tonivade.es

Records

- Introducido en Java 16.
- Inmutables.
- Constructor canónico.

commit

me@tonivade.es

Records (II)

```
public record Movie(String title, int year, int duration) {
}
```

- `equals()`
- `hashcode()`
- `toString()`

commit

me@tonivade.es

Sealed interfaces

- Introducido en Java 17.
- Jerarquías de clases cerradas.
- `non-sealed`

commit

me@tonivade.es

Sealed interfaces (II)

```
public sealed interface Shape {
  record Square(int side) implements Shape {}
  record Rectangle(int weight, int height) implements Shape {}
  record Circle(int radius) implements Shape {}
}
```

commit

me@tonivade.es

Pattern Matching for Switch

- Introducido en Java 21.
- Deconstructores, nos permiten acceder a los componentes internos.
- Exhaustiveness.

commit

me@tonivade.es

Pattern Matching for Switch (II)

```
var area = switch (this) {
    case Square(var side) -> side * side;
    case Rectangle(var weight, var height) -> weight * height;
    case Circle(var radius) -> Math.PI * Math.pow(radius, 2);
};
```

commit

me@tonivade.es

Más información

Mi charla del año pasado.

commit

me@tonivade.es

Un DSL sencillo

- Para generar queries SQL.

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  interface Table {
    String name();
  }
}
```

me@tonivade.es

commit

Un DSL sencillo

```
sealed interface SqlDsl {
  sealed interface Field<T> {}
  record Value<T>(T value) implements Field<T> {}
  record TableField<T>(String table, String name) implements Field<T> {}
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  sealed interface Filter<T> {}
  record Equal<T>(Field<T> left, Field<T> right) implements Filter<T> {}
  record GreaterThan<T>(Field<T> left, Field<T> right) implements Filter<T> {}
  record LessThan<T>(Field<T> left, Field<T> right) implements Filter<T> {}
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  sealed interface Sorting<T> {}
  record Ascending<T>(Field<T> field) implements Sorting<T> {}
  record Descending<T>(Field<T> field) implements Sorting<T> {}
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  record Query(
      List<Field<?>> fields,
      Table table,
      List<Filter<?>> filters,
      List<Sorting<?>> sorting) {
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  record Query(
      List<Field<?>> fields,
      Table table,
      List<Filter<?>> filters,
      List<Sorting<?>> sorting) {

    Query from(Table table) {
      return new Query(fields, table, filters, sorting);
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  record Query(
      List<Field<?>> fields,
      Table table,
      List<Filter<?>> filters,
      List<Sorting<?>> sorting) {

    Query where(Filter<?>... filters) {
      return new Query(fields, table, List.of(filters), sorting);
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  record Query(
      List<Field<?>> fields,
      Table table,
      List<Filter<?>> filters,
      List<Sorting<?>> sorting) {

    Query sorting(Sorting<?>... sorting) {
      return new Query(fields, table, filters, List.of(sorting));
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  static Query select(Field<?>... fields) {
    return new Query(List.of(fields), null, null, null);
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  static <T> Field<T> field(String table, String name) {
    return new TableField<>(table, name);
  }
  static <T> Field<T> value(T value) {
    return new Value<>(value);
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  static <T> Filter<T> eq(Field<T> field, T value) {
    return new Equal<>(field, value(value));
  }
  static <T> Filter<T> gt(Field<T> field, T value) {
    return new GreaterThan<>(field, value(value));
  }
  static <T> Filter<T> lt(Field<T> field, T value) {
    return new LessThan<>(field, value(value));
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  static <T> Sorting<T> asc(Field<T> field) {
    return new Ascending<>(field);
  }
  static <T> Sorting<T> desc(Field<T> field) {
    return new Descending<>(field);
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  final class People implements Table {

    final Field<String> NAME = field("people", "name");
    final Field<Integer> AGE = field("people", "age");

    public String name() {
      return "people";
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo

```
sealed interface SqlDsl {
  static void main(String... args) {
    var PEOPLE = new People();

    var query = select(PEOPLE.NAME, PEOPLE.AGE)
        .from(PEOPLE)
        .where(gt(PEOPLE.AGE, 18))
        .sorting(asc(PEOPLE.AGE));

    System.console().println(query.toSql());
  }
}
```

commit

me@tonivade.es

Un DSL sencillo (II)

```
sealed interface SqlDsl {
  record Query(...) {
    String toSql() {
      // ... a lot of boring code
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo (II)

```
sealed interface SqlDsl {
  sealed interface Field<T> {
    default String toSql() {
      return switch (this) {
        case Value(var value) -> String.valueOf(value);
        case TableField(var table, var name) -> table + "." + name;
      };
    }
  }
}
```

me@tonivade.es

commit

Un DSL sencillo (II)

```
sealed interface SqlDsl {
  sealed interface Filter<T> {
    default String toSql() {
      return switch (this) {
        case Equal(var left, var right) -> left.toSql() + " = " + right.toSql();
        case GreaterThan(var left, var right) -> left.toSql() + " > " + right.toSql();
        case LessThan(var left, var right) -> left.toSql() + " < " + right.toSql();
      };
    }
  }
}
```

commit

me@tonivade.es

Un DSL sencillo (II)

```
sealed interface SqlDsl {
  sealed interface Sorting<T> {
    default String toSql() {
      return switch (this) {
        case Ascending(var field) -> field.toSql() + " asc";
        case Descending(var field) -> field.toSql() + " desc";
      };
    }
  }
}
```

me@tonivade.es

commit

Un DSL sencillo (II)

```
select people.name, people.age from people where people.age > 18 order by people.age asc
```

commit

me@tonivade.es

Un DSL más dinámico

- Que permita:
  - escribir en pantalla.
  - leer de consola.
- Combinar estas operaciones para hacer programas.

commit

me@tonivade.es

Un programa muy sencillo

me@tonivade.es

commit

Un programa muy sencillo

```
# What's your name?
```

commit

me@tonivade.es

Un programa muy sencillo

```
# What's your name?
Toni
```

commit

me@tonivade.es

Un programa muy sencillo

```
# What's your name?
Toni
# Hello Toni!
```

commit

me@tonivade.es

Un programa muy sencillo (III)

```
sealed interface Console {
}
```

commit

me@tonivade.es

Un programa muy sencillo (III)

```
sealed interface Console {
  record WriteLine(String line) implements Console {}
}
```

commit

me@tonivade.es

Un programa muy sencillo (III)

```
sealed interface Console {
  record WriteLine(String line) implements Console {}
  record ReadLine() implements Console {}
}
```

commit

me@tonivade.es

Primer Intento

Usando CPS. Continuation Passing Style.

```
sealed interface ConsoleCps {
  record WriteLine(String line) implements ConsoleCps {}
  record ReadLine() implements ConsoleCps {}
}
```

commit

me@tonivade.es

Primer Intento

Usando CPS. Continuation Passing Style.

```
sealed interface ConsoleCps {
  record WriteLine(String line, ConsoleCps next) implements ConsoleCps {}
  record ReadLine() implements ConsoleCps {}
}
```

commit

me@tonivade.es

Primer Intento

Usando CPS. Continuation Passing Style.

```
sealed interface ConsoleCps {
  record WriteLine(String line, ConsoleCps next) implements ConsoleCps {}
  record ReadLine(Function<String, ConsoleCps> next) implements ConsoleCps {}
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?");
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?", ???);
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?",
      new ReadLine());
  }
}
```

me@tonivade.es

commit

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?",
      new ReadLine(???));
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?",
      new ReadLine(
        name -> new WriteLine("Hello " + name + "!")));
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?",
      new ReadLine(
        name -> new WriteLine("Hello " + name + "!", ???)));
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

```
sealed interface ConsoleCps {
  static void main() {
    new WriteLine("What's your name?",
      new ReadLine(
        name -> new WriteLine("Hello " + name + "!",
          new End())));
  }
}
```

commit

me@tonivade.es

Primer Intento (II)

Tenemos que añadir otro caso para terminar la ejecución:

```
sealed interface ConsoleCps {
  record WriteLine(String line, ConsoleCps next) implements ConsoleCps {}
  record ReadLine(Function<String, ConsoleCps> next) implements ConsoleCps {}
  record End() implements ConsoleCps {}
}
```

commit

me@tonivade.es

Primer Intento (III)

```
sealed interface ConsoleCps {
  static ConsoleCps prompt(String question, Function<String, ConsoleCps> next) {
    return new WriteLine(question, new ReadLine(next));
  }
}
```

commit

me@tonivade.es

Primer Intento (III)

```
sealed interface ConsoleCps {
  static void main() {
    prompt("What's your name?",
      name -> new WriteLine("Hello " + name + "!", new End()));
  }
}
```

commit

me@tonivade.es

Primer Intento (III)

```
sealed interface ConsoleCps {
  static ConsoleCps sayHello(String name) {
    return new WriteLine("Hello " + name + "!", new End());
  }
}
```

commit

me@tonivade.es

Primer Intento (III)

```
sealed interface ConsoleCps {
  static void main() {
    prompt("What's your name?", ConsoleCps::sayHello);
  }
}
```

commit

me@tonivade.es

Primer Intento (IV)

Ahora es el momento de evaluar el programa.

```
sealed interface ConsoleCps {
  static void main() {
    var program = prompt("What's your name?", ConsoleCps::sayHello);

    program.eval();
  }
}
```

commit

me@tonivade.es

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
    };
  }
}
```

me@tonivade.es

commit

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
      case WriteLine(var line, var next) -> {
        System.console().println(line);
        yield next.eval();
      }
    };
  }
}
```

commit

me@tonivade.es

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
      case WriteLine(var line, var next) -> {
        System.console().println(line);
        yield next.eval();
      }
    };
  }
}
```

commit

me@tonivade.es

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
      case WriteLine(var line, var next) -> {
        System.console().println(line);
        yield next.eval();
      }
      case ReadLine(var next) -> {
        var line = System.console().readLine();
        yield next.apply(line).eval();
      }
    };
  }
}
```

me@tonivade.es

it

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
      case WriteLine(var line, var next) -> {
        System.console().println(line);
        yield next.eval();
      }
      case ReadLine(var next) -> {
        var line = System.console().readLine();
        yield next.apply(line).eval();
      }
    };
  }
}
```

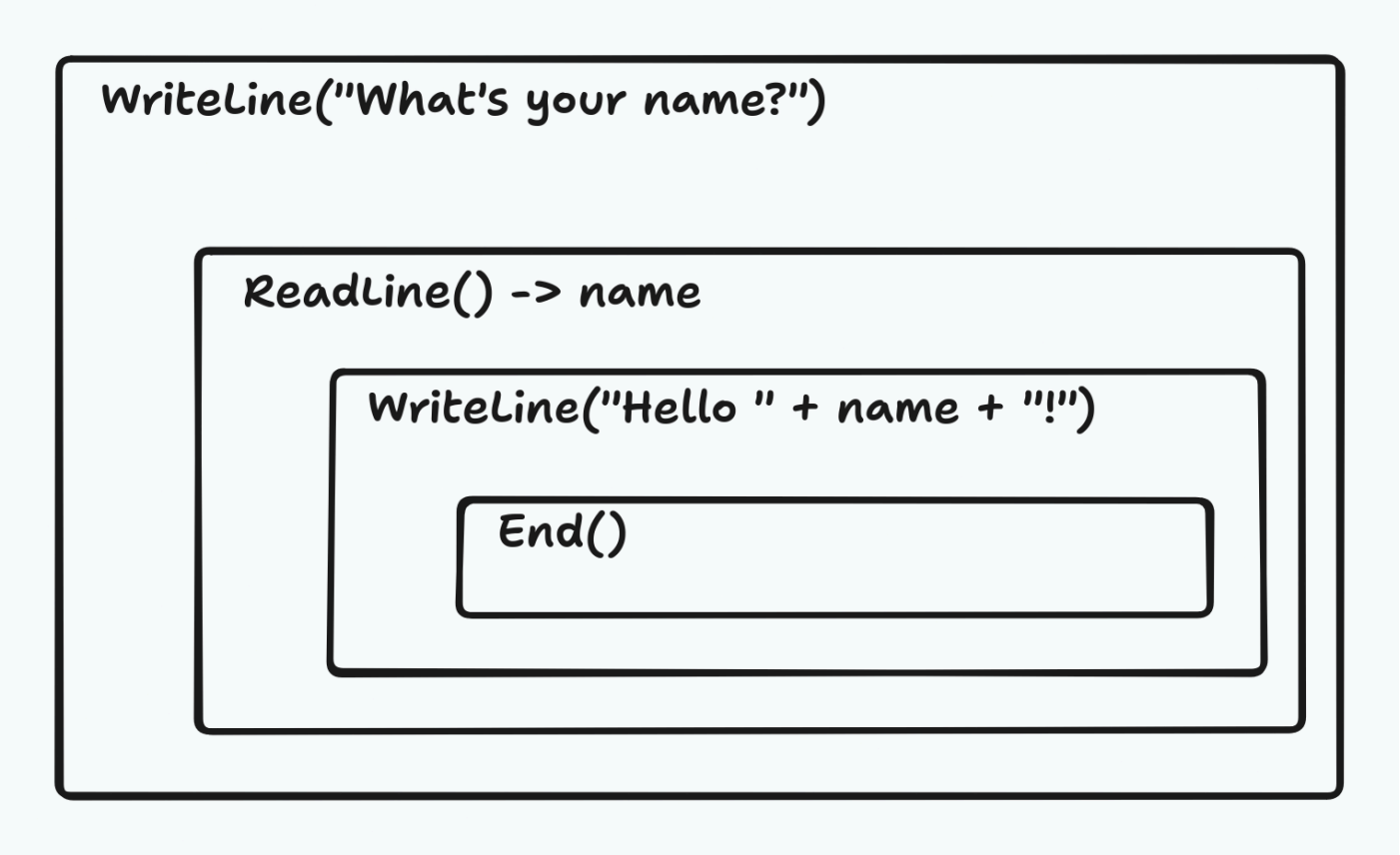
me@tonivade.es

Primer Intento (IV)

```
sealed interface ConsoleCps {
  default String eval() {
    return switch (this) {
      case WriteLine(var line, var next) -> {
        System.console().println(line);
        yield next.eval();
      }
      case ReadLine(var next) -> {
        var line = System.console().readLine();
        yield next.apply(line).eval();
      }
      case End _ -> null;
    };
  }
}
```

me@tonivade.es

Primer Intento (V)



me@tonivade.es

commit

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
}
```

commit

me@tonivade.es

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
  record WriteLine(String line) implements ConsoleDsl {}
  record ReadLine() implements ConsoleDsl {}
}
```

commit

me@tonivade.es

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
  record WriteLine(String line) implements ConsoleDsl {}
  record ReadLine() implements ConsoleDsl {}
  record AndThen() implements ConsoleDsl {}
}
```

commit

me@tonivade.es

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
  record WriteLine(String line) implements ConsoleDsl {}
  record ReadLine() implements ConsoleDsl {}
  record AndThen(???) implements ConsoleDsl {}
}
```

commit

me@tonivade.es

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
  record WriteLine(String line) implements ConsoleDsl {}
  record ReadLine() implements ConsoleDsl {}
  record AndThen(
    ConsoleDsl current)
      implements ConsoleDsl {}
}
```

commit

me@tonivade.es

Otro Intento

Usando otro estilo.

```
sealed interface ConsoleDsl {
  record WriteLine(String line) implements ConsoleDsl {}
  record ReadLine() implements ConsoleDsl {}
  record AndThen(
    ConsoleDsl current,
    Function<String, ConsoleDsl> next)
      implements ConsoleDsl {}
}
```

commit

me@tonivade.es

Otro Intento (II)

```
sealed interface ConsoleDsl {
  static void main() {
    new WriteLine("What's your name?");
  }
}
```

commit

me@tonivade.es

Otro Intento (II)

```
sealed interface ConsoleDsl {
  static void main() {
    new AndThen(
      new WriteLine("What's your name?"), _ -> new ReadLine());
  }
}
```

commit

me@tonivade.es

Otro Intento (II)

```
sealed interface ConsoleDsl {
  static void main() {
    new AndThen(
      new AndThen(
        new WriteLine("What's your name?"),
        _ -> new ReadLine()),
      name -> new WriteLine("Hello " + name + "!"));
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  default ConsoleDsl andThen(Function<String, ConsoleDsl> next) {
    return new AndThen(this, next);
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static void main() {
    new WriteLine("What's your name?");
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static void main() {
    new WriteLine("What's your name?")
      .andThen(_ -> new ReadLine());
  }
}
```

me@tonivade.es

commit

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static void main() {
    new WriteLine("What's your name?")
      .andThen(_ -> new ReadLine())
      .andThen(name -> new WriteLine("Hello " + name + "!"));
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static ConsoleDsl prompt(String question) {
    return new WriteLine(question).andThen(_ -> new ReadLine());
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static void main() {
    prompt("What's your name?")
      .andThen(name -> new WriteLine("Hello " + name + "!"));
  }
}
```

commit

me@tonivade.es

Otro Intento (III)

```
sealed interface ConsoleDsl {
  static ConsoleDsl sayHello(String name) {
    return writeLine("Hello " + name + "!");
  }
}
```

me@tonivade.es

commit

Otro Intento (IV)

```
sealed interface ConsoleDsl {
  static void main() {
    prompt("What's your name?")
      .andThen(ConsoleDsl::sayHello);
  }
}
```

commit

me@tonivade.es

Otro Intento (V)

Ahora es el momento de evaluar el programa.

```
sealed interface ConsoleDsl {
  static void main() {
    var program = prompt("What's your name?")
      .andThen(ConsoleDsl::sayHello);

    program.eval();
  }
}
```

commit

me@tonivade.es

Otro Intento (V)

```
sealed interface ConsoleDsl {
  default String eval() {
    return switch (this) {
    };
  }
}
```

commit

me@tonivade.es

Otro Intento (V)

```
sealed interface ConsoleDsl {
  default String eval() {
    return switch (this) {
      case WriteLine(var line) -> {
        System.console().println(line);
        yield null;
      }
    };
  }
}
```

commit

me@tonivade.es

Otro Intento (V)

```
sealed interface ConsoleDsl {
  default String eval() {
    return switch (this) {
      case WriteLine(var line) -> {
        System.console().println(line);
        yield null;
      }
      case ReadLine _ -> System.console().readLine();
    };
  }
}
```

commit

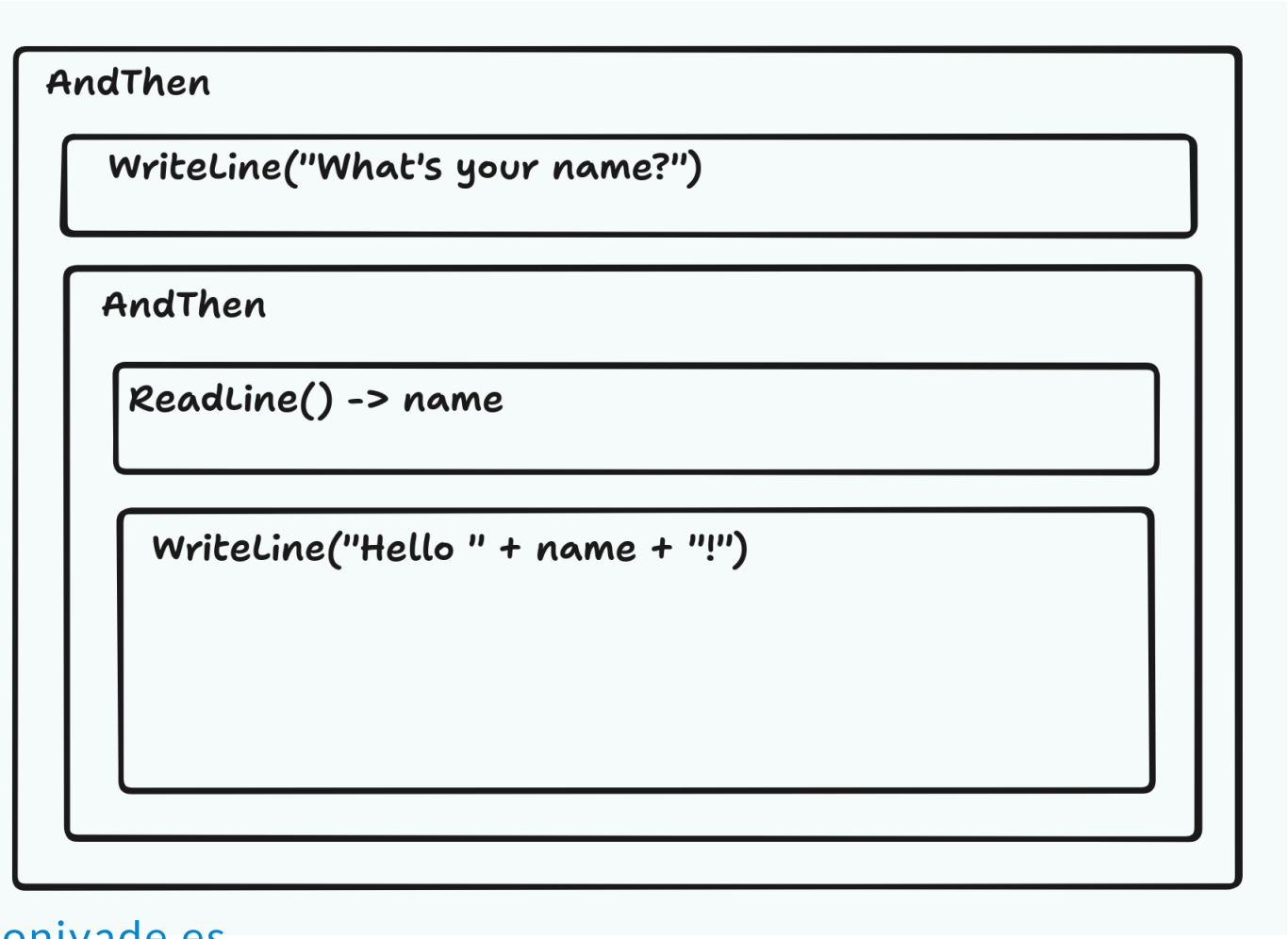
me@tonivade.es

Otro Intento (V)

```
sealed interface ConsoleDsl {
  default String eval() {
    return switch (this) {
      case WriteLine(var line) -> {
        System.console().println(line);
        yield null;
      }
      case ReadLine _ -> System.console().readLine();
      case AndThen(var current, var next)
        -> next.apply(current.eval()).eval();
    };
  }
}
```

me@tonivade.es

Otro Intento (V)



commit

me@tonivade.es

Un DSL más divertido

- Un juego sencillo.
- Adivinar un número entre 0 a 9.
- Añadir estado.

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
# Enter a number between 0 to 9
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
# Enter a number between 0 to 9
1
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
# Enter a number between 0 to 9
1
# Enter a number between 0 to 9
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
# Enter a number between 0 to 9
1
# Enter a number between 0 to 9
2
```

commit

me@tonivade.es

Un DSL más divertido

```
# Do you want to play a game? (y/n)
y
# Enter a number between 0 to 9
1
# Enter a number between 0 to 9
2
# YOU WIN!
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
}
```

me@tonivade.es

commit

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen() implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(???) implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
      GameDsl current)
          implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current, ???)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl<Void> {}
  record AndThen(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl<Void> {}
  record AndThen<T>(
    GameDsl current,
    Function<?, GameDsl> next)
      implements GameDsl<T> {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl<Void> {}
  record AndThen<T>(
    GameDsl<T> current,
    Function<?, GameDsl> next)
      implements GameDsl<T> {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl<Void> {}
  record AndThen<T>(
    GameDsl<T> current,
    Function<T, GameDsl<?>> next)
      implements GameDsl<T> {}
}
```

commit

me@tonivade.es

Un DSL más divertido (II)

```
sealed interface GameDsl<T> {
  record WriteLine(String line) implements GameDsl<Void> {}
  record ReadLine() implements GameDsl<String> {}
  record NextInt(int bound) implements GameDsl<Integer> {}
  record GetValue() implements GameDsl<Integer> {}
  record SetValue(int value) implements GameDsl<Void> {}
  record AndThen<X, T>(
    GameDsl<X> current,
    Function<X, GameDsl<T>> next)
      implements GameDsl<T> {}
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)");
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return ???;
        }
        return ???;
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return ???;
        }
        return new WriteLine("Bye!");
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return new NextInt(10);
        }
        return new WriteLine("Bye!");
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return new NextInt(10).andThen(SetValue::new);
        }
        return new WriteLine("Bye!");
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return new NextInt(10).andThen(SetValue::new).andThen(_ -> play());
        }
        return new WriteLine("Bye!");
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9");
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .andThen(number -> {
        return new GetValue();
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .andThen(number -> {
        return new GetValue()
          .andThen(value -> value == number);
      });
  }
}
```

me@tonivade.es

commit

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .andThen(number -> {
        return new GetValue()
          .andThen(value -> value == Integer.parseInt(number));
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .andThen(number -> {
        return new GetValue()
          .andThen(value -> new Done(value == Integer.parseInt(number)));
      });
  }
}
```

me@tonivade.es

commit

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  record Done<T>(T value) implements GameDsl<T> {}
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  record Done<T>(T value) implements GameDsl<T> {}

  default <R> GameDsl<R> map(Function<T, R> mapper) {
    return andThen(mapper.andThen(Done::new));
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .andThen(number -> {
        return new GetValue()
          .map(value -> value == Integer.parseInt(number));
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .map(Integer::parseInt)
      .andThen(number -> {
        return new GetValue().map(value -> value == number);
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .map(Integer::parseInt)
      .andThen(number -> new GetValue().map(value -> value == number))
      .andThen(result -> {
        if (result) {
          return ???;
        }
        return ???;
      });
  }
}
```

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .map(Integer::parseInt)
      .andThen(number -> new GetValue().map(value -> value == number))
      .andThen(result -> {
        if (result) {
          return new WriteLine("YOU WIN!");
        }
        return ???;
      });
  }
}
```

me@tonivade.es

Un DSL más divertido (III)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .map(Integer::parseInt)
      .andThen(number -> new GetValue().map(value -> value == number))
      .andThen(result -> {
        if (result) {
          return new WriteLine("YOU WIN!");
        }
        return play();
      });
  }
}
```

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  static void main() {
    var program = prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return new NextInt(10).andThen(SetValue::new).andThen(_ -> play());
        }
        return new WriteLine("Bye!");
      });

    program.eval();
  }
}
```

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default ? eval() {
    return switch (this) {
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
      case WriteLine(var line) -> {
        System.console().println(line);
        yield null;
      }
      case ReadLine _ -> System.console().readLine();
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ???
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ThreadLocalRandom.current().nextInt(bound);
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ThreadLocalRandom.current().nextInt(bound);
      case SetValue(var value) -> {
        context.set(value);
        yield null;
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval() {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ThreadLocalRandom.current().nextInt(bound);
      case SetValue(var value) -> {
        context.set(value);
        yield null;
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ThreadLocalRandom.current().nextInt(bound);
      case SetValue(var value) -> {
        context.set(value);
        yield null;
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case NextInt(var bound) -> ThreadLocalRandom.current().nextInt(bound);
      case SetValue(var value) -> {
        context.set(value);
        yield null;
      }
      case GetValue _ -> context.get();
    };
  }
}
```

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen(var current, var next) -> ???;
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, ?>(var current, var next) -> ???;
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T>(var current, var next) -> ???;
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T>(var current, var next) -> {
        current.eval(context);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T>(var current, var next) -> {
        var value = current.eval(context);
        yield next.apply(value);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T>(var current, var next) -> {
        var value = current.eval(context);
        yield next.apply(value).eval(context);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T>(var current, var next) -> {
        // looks good, but it doesn't compile 🤦
        var value = current.eval(context);
        yield next.apply(value).eval(context);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default <X> T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<X, T>(var current, var next) -> {
        var value = current.eval(context);
        yield next.apply(value).eval(context);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default <X> T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<X, T>(var current, var next) -> {
        // it doesn't work 😢
        var value = current.eval(context);
        yield next.apply(value).eval(context);
      }
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  record AndThen<X, T>(
      GameDsl<X> current,
      Function<X, GameDsl<T>> next) implements GameDsl<T> {
    private T safeEval(Context context) {
      return next.apply(current.eval(context)).eval(context);
    }
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T> andThen -> andThen.safeEval(context);
    };
  }
}
```

me@tonivade.es

commit

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  // the compiler still complains here 😢
  default T eval(Context context) {
    return switch (this) {
      // ...
      case AndThen<?, T> andThen -> andThen.safeEval(context);
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    // we have to add this cast to make the compiler happy
    return (T) switch (this) {
      // ...
      case AndThen<?, T> andThen -> andThen.safeEval(context);
    };
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (IV)

```
sealed interface GameDsl<T> {
  default T eval(Context context) {
    return (T) switch (this) {
      // ...
      case Done<T>(var value) -> value;
    };
  }
}
```

me@tonivade.es

commit

Un DSL más divertido (IV)

```
final class Context {

  private int value;

  void set(int value) { this.value = value; }

  int get() { return value; }
}
```

commit

me@tonivade.es

Un DSL más divertido (V)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> randomNumber() {
    return new NextInt(10).andThen(SetValue::new);
  }
}
```

me@tonivade.es

commit

Un DSL más divertido (V)

```
sealed interface GameDsl<T> {
  static void main() {
    prompt("Do you want to play a game? (y/n)")
      .andThen(answer -> {
        if (answer.equalsIgnoreCase("y")) {
          return randomNumber().andThen(_ -> play());
        }
        return new WriteLine("Bye!");
      });
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (V)

```
sealed interface GameDsl<T> {
  static GameDsl<Boolean> checkNumber(int number) {
    return new GetValue().map(value -> value == number);
  }
}
```

commit

me@tonivade.es

Un DSL más divertido (V)

```
sealed interface GameDsl<T> {
  static GameDsl<Void> play() {
    return prompt("Enter a number between 0 to 9")
      .map(Integer::parseInt)
      .andThen(GameDsl::checkNumber)
      .andThen(result -> {
        if (result) {
          return new WriteLine("YOU WIN!");
        }
        return play();
      });
  }
}
```

me@tonivade.es

Sacar factor común

- Esto empieza a parecerse sospechosamente a algo familiar.
- ¿o no?
- Una monada.
- Concretamente una free monad.

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program {
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  interface Dsl<T> extends Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  non-sealed interface Dsl<T> extends Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  non-sealed interface Dsl<T> extends Program<T> {
    T handle();
  }
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  default <R> Program<R> andThen(Function<T, Program<R>> next) {
    return new AndThen<>(this, next);
  }
}
```

commit

me@tonivade.es

Sacar factor común

```
sealed interface Program<T> {
  default <R> Program<R> map(Function<T, R> mapper) {
    return andThen(mapper.andThen(Done::new));
  }
}
```

commit

me@tonivade.es

Sacar factor común (II)

```
sealed interface Program<T> {
  default T eval() {
    return switch (this) {
    };
  }
}
```

me@tonivade.es

commit

Sacar factor común (II)

```
sealed interface Program<T> {
  default T eval() {
    return switch (this) {
      case Done<T>(T value) -> value;
    };
  }
}
```

me@tonivade.es

commit

Sacar factor común (II)

```
sealed interface Program<T> {
  default T eval() {
    return switch (this) {
      case Done<T>(T value) -> value;
      case AndThen<?, T> andThen -> andThen.safeEval();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (II)

```
sealed interface Program<T> {
  default T eval() {
    return switch (this) {
      case Done<T>(T value) -> value;
      case AndThen<?, T> andThen -> andThen.safeEval();
      case Dsl<T> dsl -> dsl.handle();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (III)

```
sealed interface Console<T> extends Program.Dsl<T> {
}
```

commit

me@tonivade.es

Sacar factor común (III)

```
sealed interface Console<T> extends Program.Dsl<T> {
  record WriteLine(String line) implements Console<Void> {}
  record ReadLine() implements Console<String> {}
}
```

commit

me@tonivade.es

Sacar factor común (III)

```
sealed interface Console<T> extends Program.Dsl<T> {
  default T handle() {
    return (T) switch (this) {
      case WriteLine(var line) -> {
        System.console().println(line);
        yield null;
      }
      case ReadLine _ -> System.console().readLine();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface Random<T> extends Program.Dsl<T> {
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface Random<T> extends Program.Dsl<T> {
  record NextInt(int bound) implements Random<Integer> {}
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface Random<T> extends Program.Dsl<T> {
  default T handle() {
    return (T) switch (this) {
      case NextInt(int bound) -> ThreadLocalRandom.current().nextInt(bound);
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  record SetValue(int value) implements State<Void> {}
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  record SetValue(int value) implements State<Void> {}
  record GetValue() implements State<Integer> {}
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  default T handle() {
    return (T) switch (this) {
      case SetValue(int value) -> ???;
    };
  }
}
```

me@tonivade.es

commit

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  default T handle(Context context) {
    return (T) switch (this) {
      case SetValue(int value) -> ???;
    };
  }
}
```

me@tonivade.es

commit

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  default T handle(Context context) {
    return (T) switch (this) {
      case SetValue(int value) -> {
        context.set(value);
        yield null;
      }
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (IV)

```
sealed interface State<T> extends Program.Dsl<T> {
  default T handle(Context context) {
    return (T) switch (this) {
      case SetValue(int value) -> {
        context.set(value);
        yield null;
      }
      case GetValue _ -> context.get();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  non-sealed interface Dsl<T> extends Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  record Done<T>(T value) implements Program<T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  non-sealed interface Dsl<T> extends Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  record Done<S, T>(T value) implements Program<S, T> {}
  record AndThen<X, T>(
    Program<X> current,
    Function<X, Program<T>> next)
      implements Program<T> {}
  non-sealed interface Dsl<T> extends Program<T> {}
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  record Done<S, T>(T value) implements Program<S, T> {}
  record AndThen<S, X, T>(
    Program<S, X> current,
    Function<S, X, Program<S, T>> next)
      implements Program<S, T> {}
  non-sealed interface Dsl<T> extends Program<T> {}
}
```

me@tonivade.es

commit

Sacar factor común (V)

```
sealed interface Program<S, T> {
  record Done<S, T>(T value) implements Program<S, T> {}
  record AndThen<S, X, T>(
    Program<S, X> current,
    Function<S, X, Program<S, T>> next)
      implements Program<S, T> {}
  non-sealed interface Dsl<S, T> extends Program<S, T> {}
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  record Done<S, T>(T value) implements Program<S, T> {}
  record AndThen<S, X, T>(
    Program<S, X> current,
    Function<S, X, Program<S, T>> next)
      implements Program<S, T> {}
  non-sealed interface Dsl<S, T> extends Program<S, T> {
    T handle(S state);
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  default T eval() {
    return switch (this) {
      case Done<T>(T value) -> value;
      case AndThen<?, T> andThen -> andThen.safeEval();
      case Dsl<T> dsl -> dsl.handle();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  default T eval(S state) {
    return switch (this) {
      case Done<T>(T value) -> value;
      case AndThen<?, T> andThen -> andThen.safeEval();
      case Dsl<T> dsl -> dsl.handle();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  default T eval(S state) {
    return switch (this) {
      case Done<S, T>(T value) -> value;
      case AndThen<?, T> andThen -> andThen.safeEval();
      case Dsl<T> dsl -> dsl.handle();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  default T eval(S state) {
    return switch (this) {
      case Done<S, T>(T value) -> value;
      case AndThen<S, ?, T> andThen -> andThen.safeEval(state);
      case Dsl<T> dsl -> dsl.handle();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Program<S, T> {
  default T eval(S state) {
    return switch (this) {
      case Done<S, T>(T value) -> value;
      case AndThen<S, ?, T> andThen -> andThen.safeEval(state);
      case Dsl<S, T> dsl -> dsl.handle(state);
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface State<T> extends Program.Dsl<Context, T> {
  default T handle(Context context) {
    return (T) switch (this) {
      case SetValue(int value) -> {
        context.set(value);
        yield null;
      }
      case GetValue _ -> context.get();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface State<T> extends Program.Dsl<Context, T> {
  default T handle(Context context) {
    return (T) switch (this) {
      case SetValue(int value) -> {
        context.set(value);
        yield null;
      }
      case GetValue _ -> context.get();
    };
  }
}
```

commit

me@tonivade.es

Sacar factor común (V)

```
sealed interface Console<T> extends Program.Dsl<?, T> {
  // ...
}
```

```
sealed interface Random<T> extends Program.Dsl<?, T> {
  // ...
}
```

commit

me@tonivade.es

Sacar factor común (V)

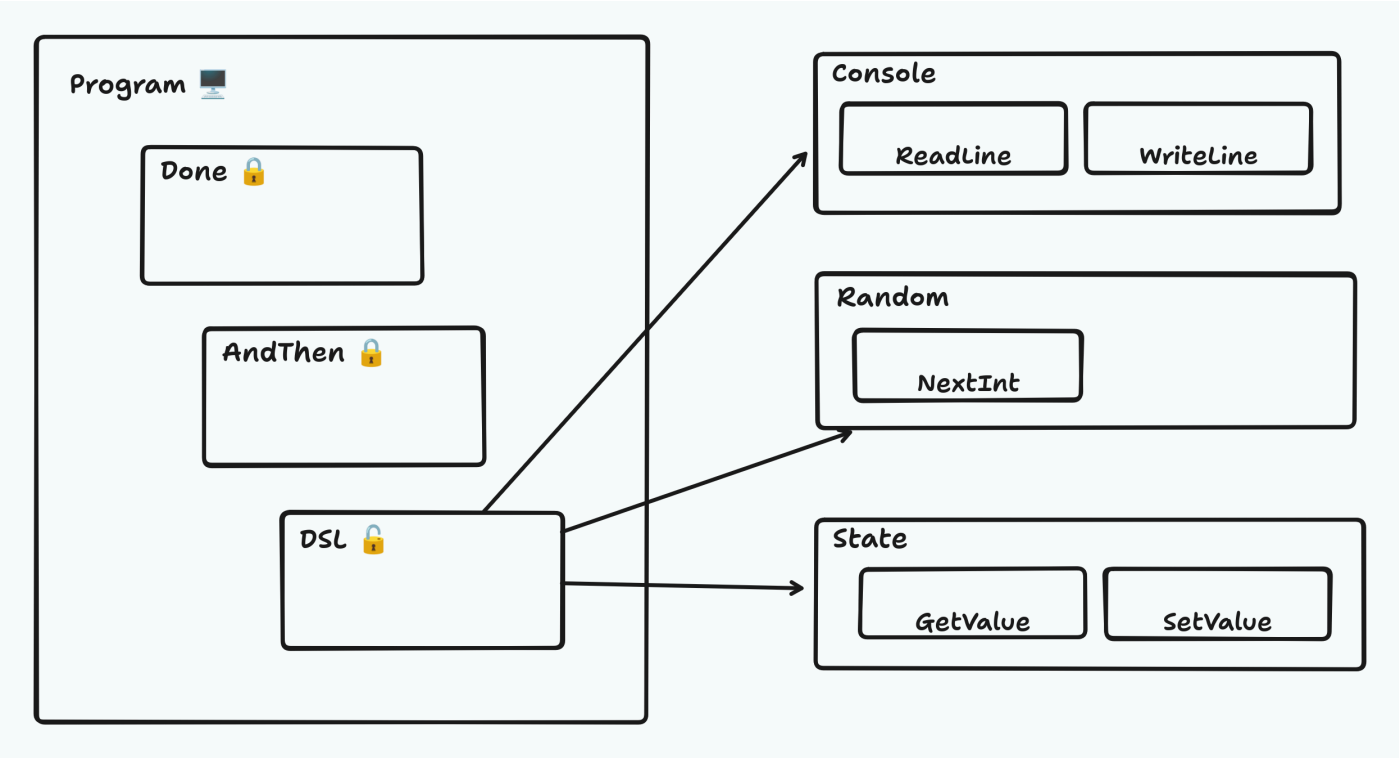
```
sealed interface Console<T> extends Program.Dsl<Void, T> {
  // ...
}
```

```
sealed interface Random<T> extends Program.Dsl<Void, T> {
  // ...
}
```

commit

me@tonivade.es

Sacar Factor Común (V)



me@tonivade.es

commit

Composición

- Todo esto es muy bonito...
- Pero hay un problema.
- La composición.

commit

me@tonivade.es

Composición

Volvamos un poco atrás

```
sealed interface GameDsl<T> {
  static GameDsl<Void> randomNumber() {
    return new NextInt(10).andThen(SetValue::new);
  }
}
```

commit

me@tonivade.es

Composición

¿Cómo se implementaría ahora?

```
class Game {
  static Program<Context, Void> randomNumber() {
    // but it doesn't compile 🔥
    return new NextInt(10).andThen(SetValue::new);
  }
}
```

commit

me@tonivade.es

Composición

```
sealed interface Random<T> extends Program.Dsl<Void, T> {
  static <S> Program<S, Integer> nextInt(int bound) {
  }
}
```

commit

me@tonivade.es

Composición

```
sealed interface Random<T> extends Program.Dsl<Void, T> {
  static <S> Program<S, Integer> nextInt(int bound) {
    // the compiler still complains 🤦
    return new NextInt(bound);
  }
}
```

commit

me@tonivade.es

Composición

```
sealed interface Random<T> extends Program.Dsl<Void, T> {
  static <S> Program<S, Integer> nextInt(int bound) {
    // now the compiler is happy ✅
    return (Program<S, Integer>) new NextInt(bound);
  }
}
```

commit

me@tonivade.es

Composición

```
class Game {
  static Program<Context, Void> randomNumber() {
    // looks good
    return Random.nextInt(10).andThen(SetValue::new);
  }
}
```

commit

me@tonivade.es

Composición

```
class Game {
  static Program<Context, Void> randomNumber() {
    // but it doesn't compile yet
    return Random.nextInt(10).andThen(SetValue::new);
  }
}
```

commit

me@tonivade.es

Composición

```
class Game {
  static Program<Context, Void> randomNumber() {
    // now it works 🥳
    return Random.<Context>nextInt(10).andThen(SetValue::new);
  }
}
```

commit

me@tonivade.es

Composición

```
class Game {
  static Program<Context, Boolean> checkNumber(int number) {
    return State.<Context>getValue().map(value -> value == number);
  }
}
```

commit

me@tonivade.es

Composición

```
sealed interface Console<T> extends Program.Dsl<Void, T> {
  static <S> Program<S, Void> writeLine(String line) {
    return (Program<S, Void>) new WriteLine(line);
  }
  static <S> Program<S, String> readLine() {
    return (Program<S, String>) new ReadLine();
  }
}
```

me@tonivade.es

commit

Composición

```
sealed interface Console<T> extends Program.Dsl<Void, T> {
  static <S> Program<S, String> prompt(String question) {
    return Console.<S>writeLine(question).andThen(readLine());
  }
}
```

me@tonivade.es

commit

Composición

```
class Game {
  static void main() {
    var program = Console.<Context>prompt("Do you want to play a game? (y/n)")
        .andThen(answer -> {
          if (answer.equalsIgnoreCase("y")) {
            return randomNumber().andThen(play());
          }
          return writeLine("Bye!");
        });

    program.eval(new Context());
  }
}
```

me@tonivade.es

Composición

```
class Game {
  static Program<Context, Void> play() {
    return Console.<Context>prompt("Enter a number between 0 and 9")
      .map(Integer::parseInt)
      .andThen(Game::checkNumber)
      .andThen(result -> {
        if (result) {
          return writeLine("YOU WIN!!");
        }
        return play();
      });
  }
}
```

me@tonivade.es

Composición

- Es necesario para facilitar la composición.
- Que es de lo que se trata todo esto.
- Componer programas a partir de otros mini programas.

commit

me@tonivade.es

Resumiendo ✍️

- Parte común.
- Que se puede extender definiendo pequeños mini lenguajes.
- Resulta complejo la composición de diferentes mini lenguajes.

commit

me@tonivade.es

Resumiendo ✍️

Cada mini lenguaje define:

- un contexto (opcional).
- un conjunto de operaciones.
- una sintaxis:
  - para definir otras operaciones y extender ese mini lenguaje.
  - poder componer ese lenguaje con otros mini lenguajes.

commit

me@tonivade.es

Conclusiones

- Definir diferentes DSLs y usarlos dentro de un mismo programa.
- Definir pequeños mini programas.
- Componer programas mayores a partir de estos mini programas.
- Building blocks 

commit

me@tonivade.es

Qué falta? 🤕

- Tail recursion.
- Mejor inferencia de tipos.

commit

me@tonivade.es

Voy a hablar de mi libro 📖

- A raíz de esto he desarrollado una librería que implementa esto.
- Con algunas mejoras:
  - Generación automática de código repetitivo usando procesadores de anotaciones.
  - Gestión de errores.
  - Structured Concurrency.
  - Retry and Repeat.
- Si tenéis interés esta en github y se llama diesel.

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
interface Console {
  void writeLine(String);
  String readLine();
}
```

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
@Diesel
interface Console {
  void writeLine(String);
  String readLine();
}
```

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
sealed interface ConsoleDsl<T> extends Program.Dsl<Console, T> {
  record WriteLine(String line) implements ConsoleDsl<Void> {}
  record ReadLine() implements ConsoleDsl<String> {}
  static <S> Program<S, Void> writeLine(String line) {
    return (Program<S, Void>) new WriteLine(line);
  }
  static <S> Program<S, String> readLine() {
    return (Program<S, String>) new ReadLine();
  }
}
```

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
sealed interface ConsoleDsl<T> extends Program.Dsl<Console, T> {
  // ...
  default T eval(Console console) {
    return (T) switch (this) {
      case WriteLine(var line) -> {
        console.writeLine(line);
        yield null;
      }
      case ReadLine() -> console.readLine();
    };
  }
}
```

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
@Diesel
interface Console {
  static void main() {
    var program = ConsoleDsl.writeLine("What's your name?")
      .andThen(_ -> ConsoleDsl.readLine())
      .andThen(name -> ConsoleDsl.writeLine("Hello " + name + "!"));
  }
}
```

commit

me@tonivade.es

Voy a hablar de mi libro 📖

```
@Diesel
interface Console {
  static void main() {
    // ...
    program.eval(new Console() {
      public void writeLine(String line) {
        System.console().println(line);
      }
      public String readLine() {
        System.console().readLine();
      }
    });
  }
}
```

me@tonivade.es

Voy a hablar de mi libro 📖

```
@Diesel
interface Console {
  static void main() {
    // ...
    program.eval(new Console() {
      public void writeLine(String line) {
        System.console().println(line);
      }
      public String readLine() {
        System.console().readLine();
      }
    });
  }
}
```

me@tonivade.es

Voy a hablar de mi libro 📖

commit

me@tonivade.es

¿Preguntas? 🤔

commit

me@tonivade.es

¡Gracias! 

commit

me@tonivade.es

Feedback 🔄

me@tonivade.es

commit

Links 🔗

- código fuente
- diesel

commit

me@tonivade.es