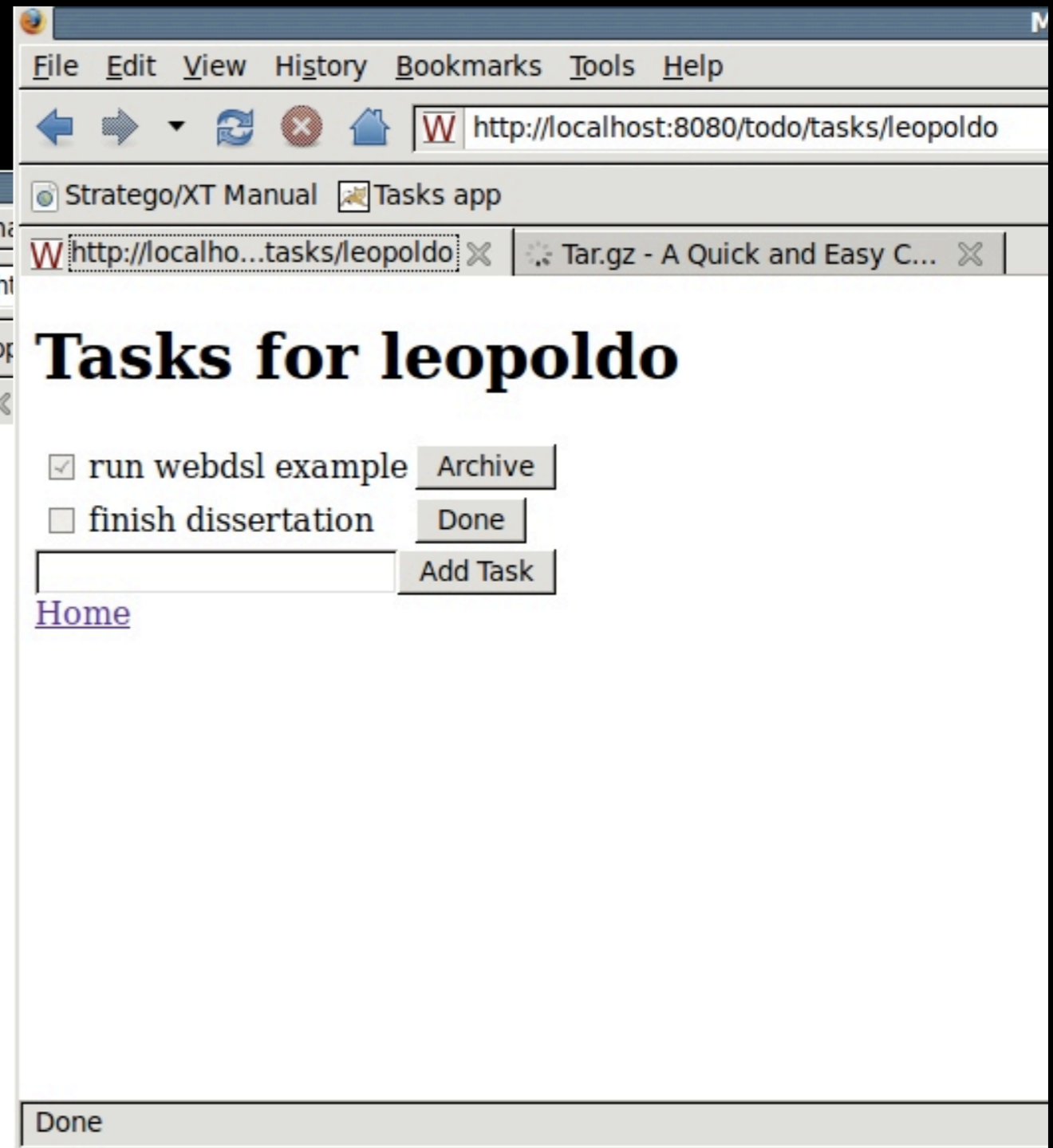
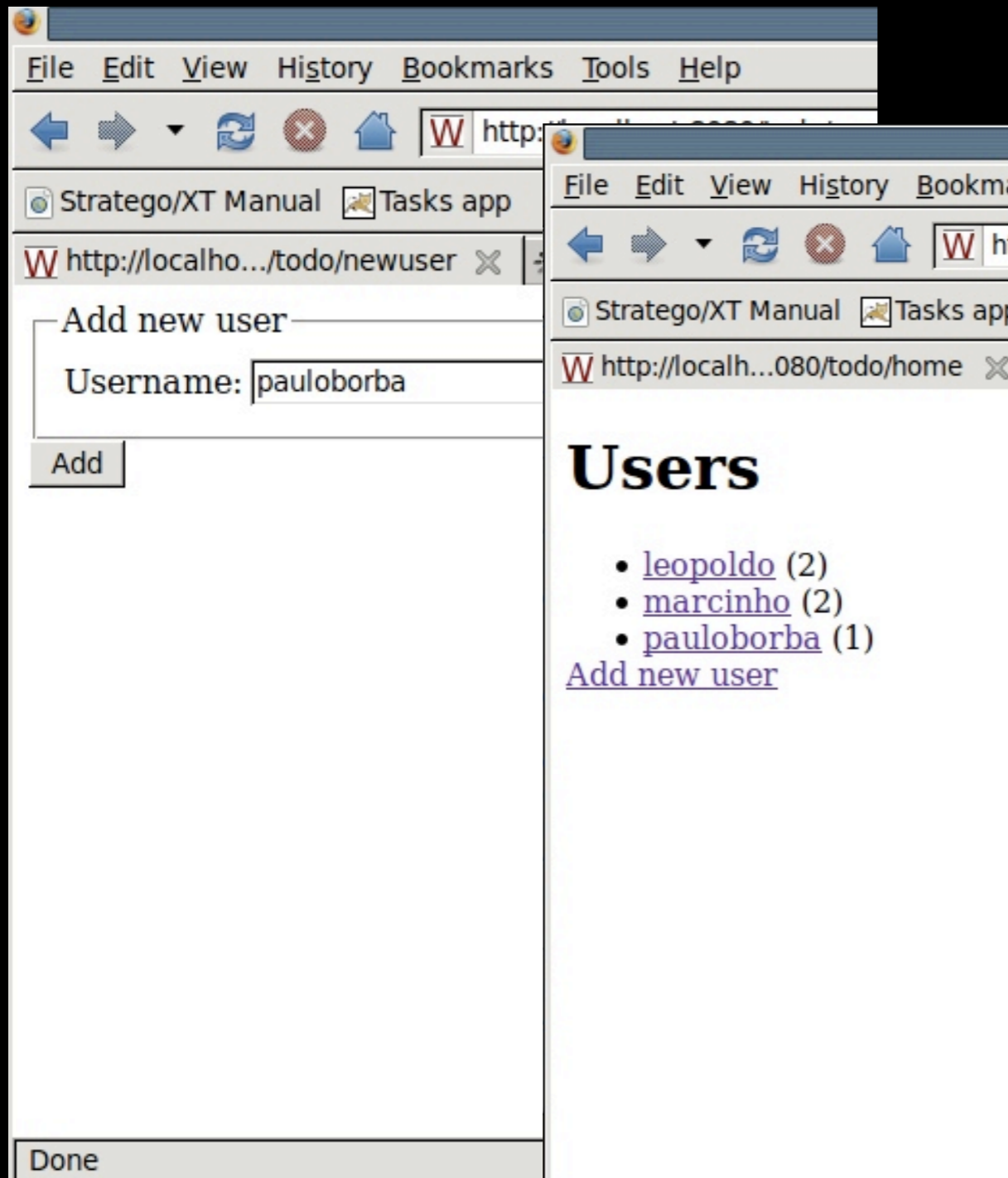


Software Productivity

Paulo Borba
Informatics Center
Federal University of Pernambuco

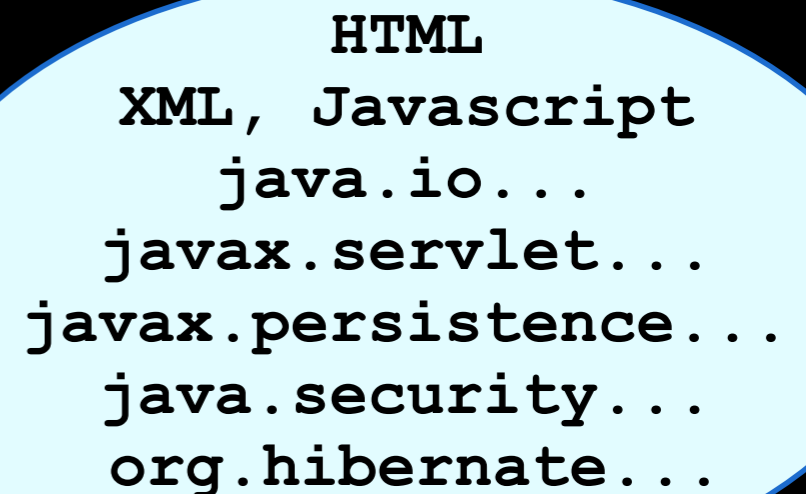
phmb@cin.ufpe.br ♦ twitter.com/pauloborba

Task lists on the web



This is a simple, typical, web application with more than...

- 20 (100) files
- 10 (20) types
- 500 (7000) LOC
- 5 (10) APIs and technologies
- ? bugs



HTML
XML, Javascript
java.io...
javax.servlet...
javax.persistence...
java.security...
org.hibernate...

Actually, 80 lines of code!

```
define page home() {
  section{
    header{"Users"}
    list{
      for(user : User) {
        listitem{
          navigate(tasks(user))
            {output(user.username)}
            (" output(user.tasks.length) ")
        }
      }
    }
    navigate(newuser()) {"Add new user"}
  }
}
```

```
application todo
  entity User {
    username :: String (id,name)
    tasks    -> List<Task>
    log      -> List<Task>
  }
  entity Task {
    description :: String
    done        :: Bool
  }
```

```

application todo
entity User {
  username :: String (id,name)
  tasks    -> List<Task>
  log      -> List<Task>
}
entity Task {
  description :: String
  done        :: Bool
}

```

```

define page home() {
  section{
    header{"Users"}
    list{
      for(user : User) {
        listitem{
          navigate(tasks(user))
            {output(user.username)}
            " (" output(user.tasks.length) )"
        }
      }
    }
    navigate(newuser()) {"Add new user"}
  }
}

```

```

define page newuser() {
  var user : User := User {}
  form{
    group("Add new user") {
      derive editRows from user for ( username )
    }
    action("Add", add())
    action add() {
      user.save();
      return tasks(user);
    }
  }
}

```

```

define page tasks(user : User) {
  var newTask : Task := Task{ done := false }
  action addtask() {
    user.tasks.add(newTask);
    newTask.save();
  }
  action done(task : Task) {
    task.done := true;
    task.save();
  }
  action archive(task : Task) {
    user.tasks.remove(task);
    user.log.add(task);
    user.save();
  }
  section{
    header{"Tasks for " output(user.username) }
    table{
      for(task : Task in user.tasks) {
        row{
          output(task.done)
          output(task.description)
          form{
            if(!task.done) {
              action("Done", done(task))
            } else {
              action("Archive", archive(task))
            }
          }
        }
      }
    }
    form{
      input(newTask.description)
      action("Add Task", addtask())
    }
    navigate(home()) {"Home"}
  }
}

```

Transformations!

```
application todo
entity User {
  username : String (id.name)
  tasks -> List<Task>
  log -> List<Task>
}
entity Task {
  description : String
  done : Bool
}

define page home() {
  section()
  header("Users")
  list()
  for (user : User) {
    listitem()
    navigate(tasks(user))
    output(user.username)
    (" output(user.tasks.length) ")
  }
  navigate(newuser()) {"Add new user"}
}

define page newuser() {
  var user : User := User ()
  group("Add new user") {
    derive editform from user for ( username )
  }
  action("Add", add())
  action("Add", add()) {
    user.save()
    return tasks(user)
  }
}

define page home2(user : User) {
  var newTask : Task := Task( done := false )
  action addtask() {
    user.tasks.add(newTask);
    newTask.save();
  }
  action done(task : Task) {
    task.done = true;
    task.save();
  }
  section()
  header("Tasks for " output(user.username) )
  table()
  for (task : Task in user.tasks) {
    row()
    output(task.done)
    output(task.description)
    form()
    if (task.done) {
      action("Done", done(task))
    } else {
      action("Archive", archive(task))
    }
  }
}

define page archive(task : Task) {
  user.tasks.remove(task);
  user.save();
}

define page archive2(task : Task) {
  user.save();
}

define page archive3(task : Task) {
  user.save();
}

define page archive4(task : Task) {
  user.save();
}

define page archive5(task : Task) {
  user.save();
}

define page archive6(task : Task) {
  user.save();
}

define page archive7(task : Task) {
  user.save();
}

define page archive8(task : Task) {
  user.save();
}

define page archive9(task : Task) {
  user.save();
}

define page archive10(task : Task) {
  user.save();
}

define page archive11(task : Task) {
  user.save();
}

define page archive12(task : Task) {
  user.save();
}

define page archive13(task : Task) {
  user.save();
}

define page archive14(task : Task) {
  user.save();
}

define page archive15(task : Task) {
  user.save();
}

define page archive16(task : Task) {
  user.save();
}

define page archive17(task : Task) {
  user.save();
}

define page archive18(task : Task) {
  user.save();
}

define page archive19(task : Task) {
  user.save();
}

define page archive20(task : Task) {
  user.save();
}

define page archive21(task : Task) {
  user.save();
}

define page archive22(task : Task) {
  user.save();
}

define page archive23(task : Task) {
  user.save();
}

define page archive24(task : Task) {
  user.save();
}

define page archive25(task : Task) {
  user.save();
}

define page archive26(task : Task) {
  user.save();
}

define page archive27(task : Task) {
  user.save();
}

define page archive28(task : Task) {
  user.save();
}

define page archive29(task : Task) {
  user.save();
}

define page archive30(task : Task) {
  user.save();
}

define page archive31(task : Task) {
  user.save();
}

define page archive32(task : Task) {
  user.save();
}

define page archive33(task : Task) {
  user.save();
}

define page archive34(task : Task) {
  user.save();
}

define page archive35(task : Task) {
  user.save();
}

define page archive36(task : Task) {
  user.save();
}

define page archive37(task : Task) {
  user.save();
}

define page archive38(task : Task) {
  user.save();
}

define page archive39(task : Task) {
  user.save();
}

define page archive40(task : Task) {
  user.save();
}

define page archive41(task : Task) {
  user.save();
}

define page archive42(task : Task) {
  user.save();
}

define page archive43(task : Task) {
  user.save();
}

define page archive44(task : Task) {
  user.save();
}

define page archive45(task : Task) {
  user.save();
}

define page archive46(task : Task) {
  user.save();
}

define page archive47(task : Task) {
  user.save();
}

define page archive48(task : Task) {
  user.save();
}

define page archive49(task : Task) {
  user.save();
}

define page archive50(task : Task) {
  user.save();
}

define page archive51(task : Task) {
  user.save();
}

define page archive52(task : Task) {
  user.save();
}

define page archive53(task : Task) {
  user.save();
}

define page archive54(task : Task) {
  user.save();
}

define page archive55(task : Task) {
  user.save();
}

define page archive56(task : Task) {
  user.save();
}

define page archive57(task : Task) {
  user.save();
}

define page archive58(task : Task) {
  user.save();
}

define page archive59(task : Task) {
  user.save();
}

define page archive60(task : Task) {
  user.save();
}

define page archive61(task : Task) {
  user.save();
}

define page archive62(task : Task) {
  user.save();
}

define page archive63(task : Task) {
  user.save();
}

define page archive64(task : Task) {
  user.save();
}

define page archive65(task : Task) {
  user.save();
}

define page archive66(task : Task) {
  user.save();
}

define page archive67(task : Task) {
  user.save();
}

define page archive68(task : Task) {
  user.save();
}

define page archive69(task : Task) {
  user.save();
}

define page archive70(task : Task) {
  user.save();
}

define page archive71(task : Task) {
  user.save();
}

define page archive72(task : Task) {
  user.save();
}

define page archive73(task : Task) {
  user.save();
}

define page archive74(task : Task) {
  user.save();
}

define page archive75(task : Task) {
  user.save();
}

define page archive76(task : Task) {
  user.save();
}

define page archive77(task : Task) {
  user.save();
}

define page archive78(task : Task) {
  user.save();
}

define page archive79(task : Task) {
  user.save();
}

define page archive80(task : Task) {
  user.save();
}

define page archive81(task : Task) {
  user.save();
}

define page archive82(task : Task) {
  user.save();
}

define page archive83(task : Task) {
  user.save();
}

define page archive84(task : Task) {
  user.save();
}

define page archive85(task : Task) {
  user.save();
}

define page archive86(task : Task) {
  user.save();
}

define page archive87(task : Task) {
  user.save();
}

define page archive88(task : Task) {
  user.save();
}

define page archive89(task : Task) {
  user.save();
}

define page archive90(task : Task) {
  user.save();
}

define page archive91(task : Task) {
  user.save();
}

define page archive92(task : Task) {
  user.save();
}

define page archive93(task : Task) {
  user.save();
}

define page archive94(task : Task) {
  user.save();
}

define page archive95(task : Task) {
  user.save();
}

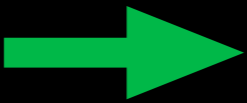
define page archive96(task : Task) {
  user.save();
}

define page archive97(task : Task) {
  user.save();
}

define page archive98(task : Task) {
  user.save();
}

define page archive99(task : Task) {
  user.save();
}

define page archive100(task : Task) {
  user.save();
}
```



```
build.xml
1 <project name="todo" default="compile">
2
3 <property name="hibernate.c3p0.acquire_increment" value="1"/>
4 <property name="hibernate.c3p0.min_size" value="0"/>
5 <property name="hibernate.c3p0.max_size" value="0"/>
6 <property name="hibernate.c3p0.timeout" value="0"/>
7 <property name="hibernate.c3p0.max_statements" value="0"/>
8 <property name="hibernate.c3p0.idle_test_period" value="0"/>
9 <property name="hibernate.c3p0.preferred_connection_driver" value="org.hibernate.connection.DriverDelegate"/>
10 </project>

hibernate.properties
hibernate.c3p0.acquire_increment=1
hibernate.c3p0.min_size=0
hibernate.c3p0.max_size=0
hibernate.c3p0.timeout=0
hibernate.c3p0.max_statements=0
hibernate.c3p0.idle_test_period=0
hibernate.c3p0.preferred_connection_driver=org.hibernate.connection.DriverDelegate

User.java
package todo.domain;

import java.util.*;
import javax.persistence.*;
import java.io.Serializable;
import utils.*;
import java.io.PrintWriter;
import org.webds1.lang.Environment;

@Entity
@Table(name = "_User", uniqueConstraints = {
    @UniqueConstraint(columnNames = {"username"})
})
public class User {
    private String username;
    private List<Task> tasks;
    private List<Task> log;

    public User() {}
    public User(String username) {
        this.username = username;
    }
    public User(String username, List<Task> tasks, List<Task> log) {
        this.username = username;
        this.tasks = tasks;
        this.log = log;
    }

    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public List<Task> getTasks() {
        return tasks;
    }
    public void setTasks(List<Task> tasks) {
        this.tasks = tasks;
    }
    public List<Task> getLog() {
        return log;
    }
    public void setLog(List<Task> log) {
        this.log = log;
    }
}

addtask1.java
package todo.actions;

import todo.domain.*;

public interface addtask1 {
    public void addTask(User user, Task task);
}

HomePage.java
package todo.actions;

import java.io.PrintWriter;
import org.webds1.lang.Environment;
import utils.TemplateCall;

public class HomePage {
    public void render(PrintWriter sout, Environment env) {
        sout.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" \"http://www.w3.org/TR/html4/strict.dtd\"");
        sout.println("<html>");
        sout.println("<head>");
        sout.println("<title>" + getPageTitle() + "</title>");
        sout.println("<link href=\"" + env.get("css") + "\" rel=\"stylesheet\" type=\"text/css\"/>");
        sout.println("<link href=\"" + env.get("js") + "\" rel=\"script\" type=\"text/javascript\"/>");
        sout.println("<script src=\"" + env.get("js") + "\" type=\"text/javascript\"/>");
        sout.println("</script>");
        sout.println("</head>");
        sout.println("<body>");
        renderIncomingSuccessMessages(sout);
        s.flush();
    }
}

EntityType.java
package utils;

public class EntityType {
    public static Object loadEntity(org.hibernate.Session hibSession, String c, String s) {
        try {
            return hibSession.load(c, s);
        } catch (Exception e) {
            return null;
        }
    }
}
```

Productivity

and

Quality

[http://twiki.cin.ufpe.br/
twiki/bin/view/TAES/
ProdutividadeGraduacao](http://twiki.cin.ufpe.br/twiki/bin/view/TAES/ProdutividadeGraduacao)
201001

Teaching
Teaching &
Understanding
Understanding

Intended learning outcome

- Develop web applications using the WebDSL
- Identify and justify necessary changes to a DSL
- Define, implement and change DSLs

Intended learning outcome

- Analyze and explain benefits and drawbacks of developing web applications with
 - general purpose languages and frameworks
 - DSL
- Develop a business case for a DSL

Applications

- Research group management system
- Technical colaborators management system

Introduce yourself...

- Name
- What do you expect from this course?

Choose your
group, now!

Software Productivity

Paulo Borba
Informatics Center
Federal University of Pernambuco

phmb@cin.ufpe.br ♦ twitter.com/pauloborba