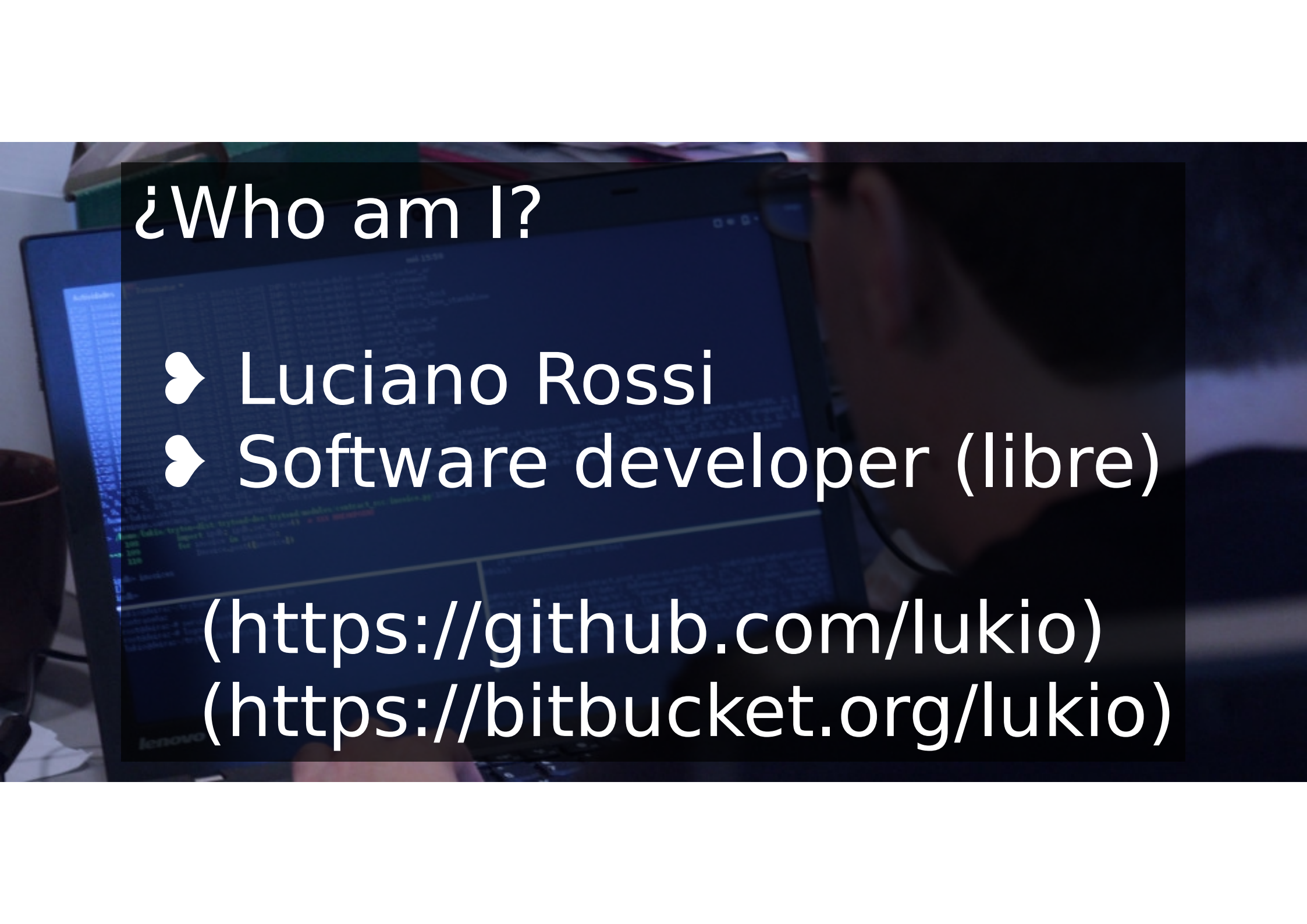




¡Bienvenidos!

Tryton



¿Who am I?

- Luciano Rossi
- Software developer (libre)

(<https://github.com/lukio>)

(<https://bitbucket.org/lukio>)

Ayni

- ▶ Iukio
- ▶ Cooperativa de Trabajo

(Video: <http://gcoop.coop/codigos-cooperativos>)

Pulsa una tecla para continuar



Plataforma de desarrollo de aplicaciones
de alto nivel Arquitectura de 3 Capas.

- * GPLv3
- * Cliente PyGtk / Cliente Web (sao)
Ligero (No tiene logica de negocios)
- * Servidor Python
- * Postgresql / SQLite / MySQL

A background image showing a person from the back, wearing a dark blue t-shirt with the word 'Tryton' printed on the back in a light color. The person is standing outdoors, and their hands are visible, holding a small object. The background is slightly blurred, showing some greenery and a wall.

Objetivos de Tryton

Consistencia

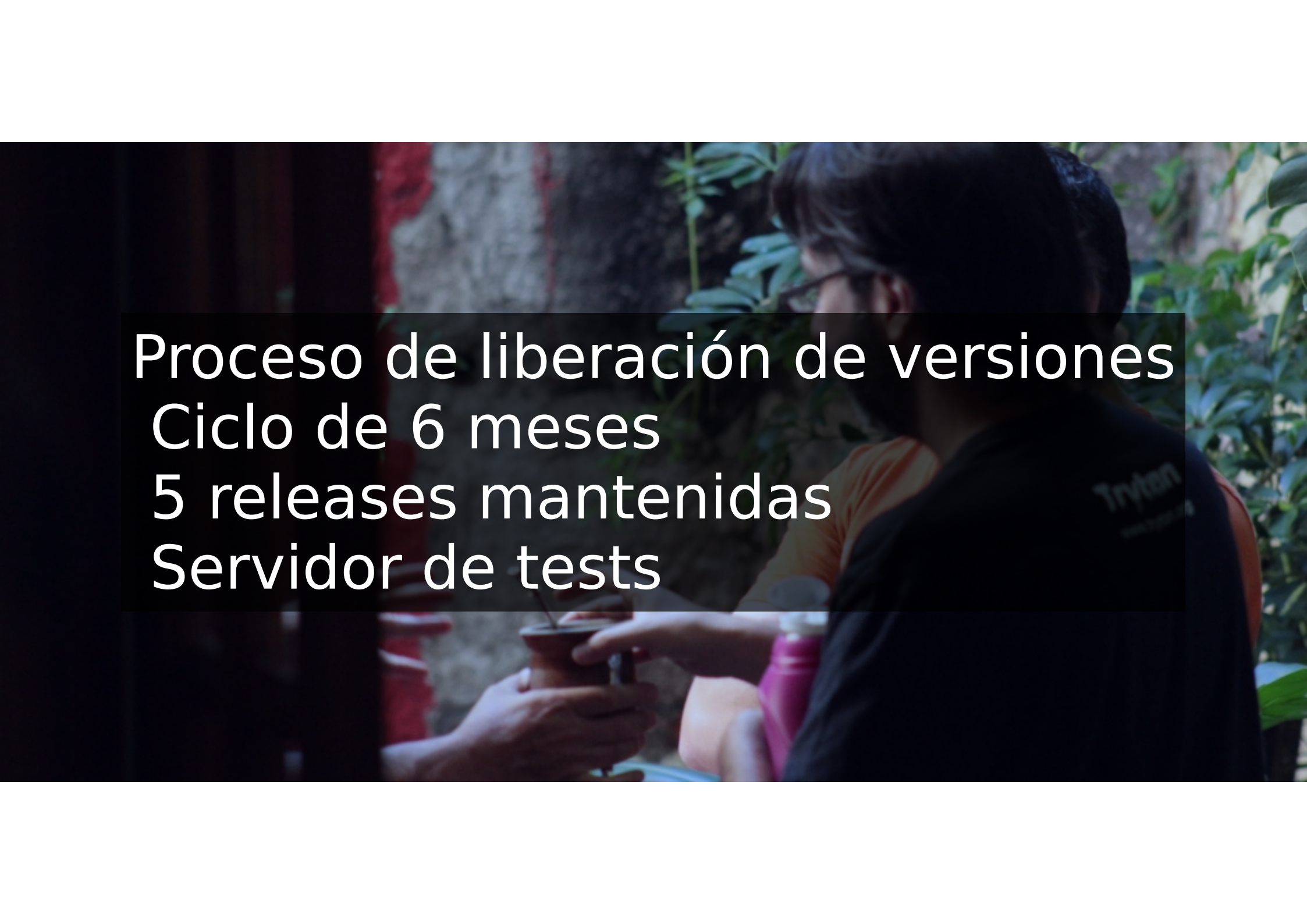
Migración

Modularidad

Enfoque “Framework especializado”

Paquetes Python → pip

Code review

A background image showing a person from the back, wearing a dark shirt with 'Tryton' on the sleeve, holding a traditional mate cup. The scene is outdoors with green foliage in the background.

Proceso de liberación de versiones
Ciclo de 6 meses
5 releases mantenidas
Servidor de tests

A person wearing a dark jacket and glasses is looking down at a device in their hands. The background is filled with green foliage, suggesting an outdoor setting. The image is dark and serves as a background for the text.

Servidor Python

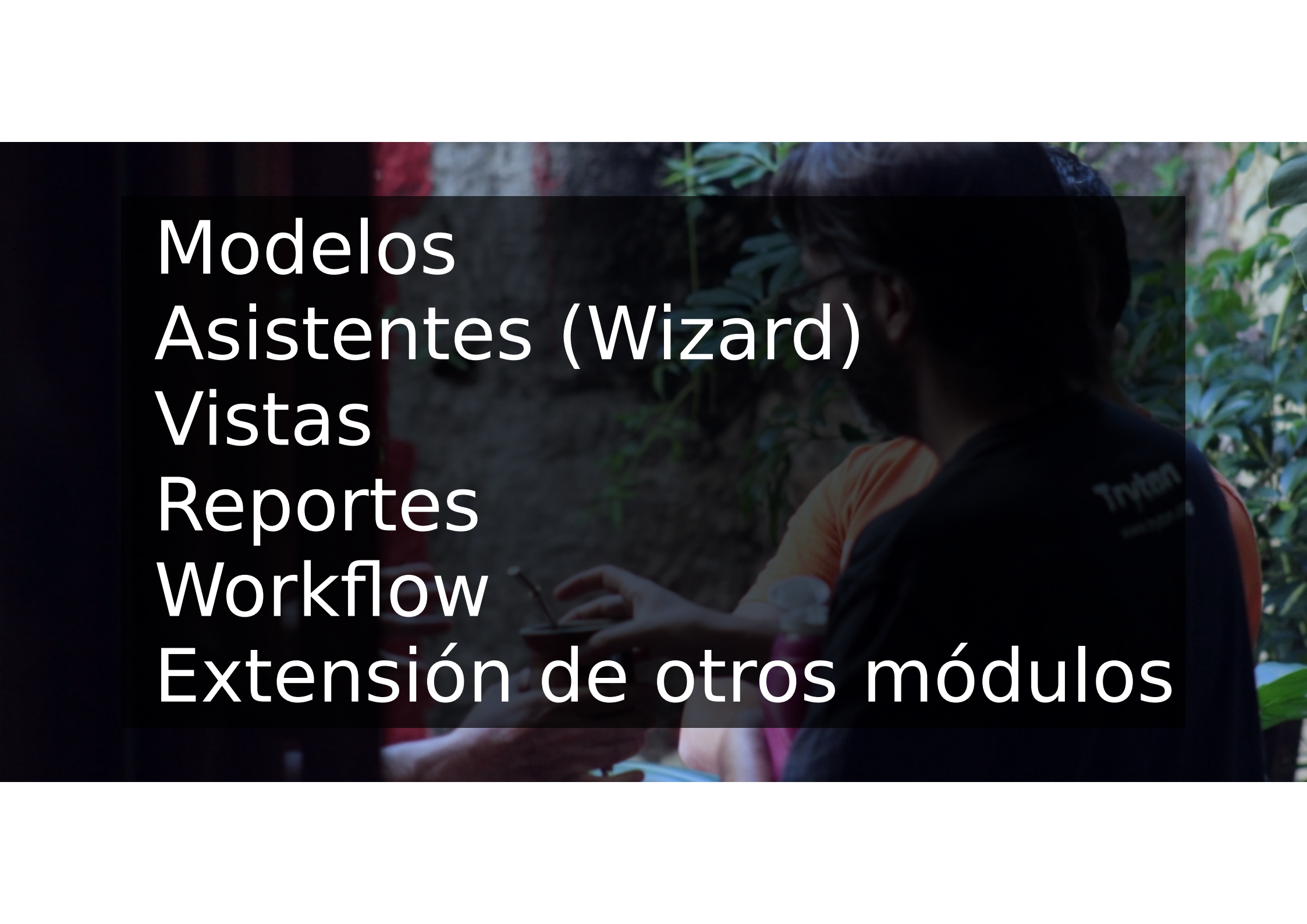
ORM

Creación de esquema
Migración automática

Workflows

Reportes (relatorio)

Multi-lenguajes



Modelos
Asistentes (Wizard)
Vistas
Reportes
Workflow
Extensión de otros módulos

A background image showing a person from the side, wearing a dark shirt, holding a traditional mate cup and drinking. The scene is outdoors with green foliage in the background. The image is dimmed to serve as a background for the text.

Modelos

Modelos

Campos

Valores por defecto

Search – Create – Write – Delete

Browse

Vistas

Estados

```
1 from trytond.model import ModelSQL
2
3 class Event (ModelSQL):
4     'Event'
5     __name__ = 'calendar.event'
```



```
1 class Event (ModelSQL, ModelView):
2     'Event '
3     __name__ = 'calendar.event'
4
5     calendar = fields.Many2One('calendar.calendar', 'Calendar',
6                                required=True)
7     summary = fields.Char('Summary')
8     description = fields.Text('Description')
9     dtstart = fields.DateTime('Start Date', required=True)
10    classification = fields.Selection([
11        ('public', 'Public'),
12        ('private', 'Private'),
13    ], 'Classification', required=True)
```

```
1 class Event (ModelSQL, ModelView):
2
3     classification = fields.Selection([
4         ('public', 'Public'),
5         ('private', 'Private'),
6     ], 'Classification', required=True)
7
8     @staticmethod
9     def default_classification():
10         return 'public'
```



```
1  calendar = Calendar.search([
2      ('owner.email', '=', 'juan@mycompany.com'),
3      ])
4
5  event = Event.create({
6      'summary': 'Reunion',
7      'calendar': calendar.id,
8      })
9
10 Event.write(event.id, {
11     'description': 'Planificar charla PyConAr',
12     })
13
14 Event.delete(event.id)
```

```
1 >>> event = Event(event_id)
2
3 >>> print event.dtstart
4 datetime.date(2012, 11, 7)
5
6 >>> for attendee in event.attendees:
7     ...     print attendee.name
8 Han Solo
9 Luke Skywalker
```



```
1 <record model="ir.ui.view" id="event_view_tree">
2   <field name="model">calendar.event</field>
3   <field name="type">tree</field>
4   <field name="arch" type="xml">
5     <![CDATA[
6       <tree string="Events">
7         <field name="calendar"/>
8         <field name="summary"/>
9         <field name="dtstart"/>
10        <field name="dtend"/>
11      </tree>
12    ]]>
13   </field>
14 </record>
```

```
1 <record model="ir.ui.view" id="event_view_form">
2   <field name="model">calendar.event</field>
3   <field name="type">form</field>
4   <field name="arch" type="xml">
5     <![CDATA[
6       <form string="Event">
7         <label name="summary"/>
8         <field name="summary"/>
9         <separator name="description" colspan="4"/>
10        <field name="description" colspan="4"/>
11      </form>
12    ]]>
13   </field>
14 </record>
```

```
1 from trytond.pyson import If, Bool, Eval
2
3 class Event(ModelSQL, ModelView):
4
5     attendees = fields.One2Many('calendar.event.attendee', 'event',
6                                 'Attendees')
7     organizer = fields.Char('Organizer', states={
8         'required': If(Bool(Eval('attendees')), ~Eval('parent'), False),
9         }, depends=['attendees'])
```


A background image showing a person from the side, wearing a dark blue shirt with the 'Tryton' logo on the sleeve. They are holding a small dark cup with a metal straw. The background is slightly blurred, showing some greenery and a wall.

Proteus

Librería para acceder a los modelos
de Tryton como un cliente

CLI Python

1 Creando un Party

2 *no no no no no no no no no no no no no no no no*

3

4 Primero instanciamos un nuevo Party:

5

6 >>> Party = Model.get('party.party')

7 >>> party = Party()

8 >>> party.id < 0

9 True

10

11 Llenamos los campos:

12

13 >>> party.name = 'ham'

14

15 Grabamos la instancia en el servidor:

16

17 >>> party.save()

18 >>> party.name

19 u'ham'

20 >>> party.id > 0

21 True





galatea (flask_tryton)

Sitio web en Flask que utiliza a Tryton como backend para armar una tienda virtual

https://bitbucket.org/zikzakmedia/tryton_galatea

A background image showing a person from the back, wearing a dark blue shirt with the 'Tryton' logo and 'www.tryton.org' on the sleeve. They are standing outdoors near some green plants. A semi-transparent dark rectangle is overlaid on the left side of the image, containing white text.

Contribuir en Tryton

Testear

Realizar codereview

Subir parches

A background image showing a person from the back, wearing a dark shirt with 'Tryton' written on it, standing outdoors near some greenery. The image is dimly lit and serves as a backdrop for the text.

¿Dónde consultar?

Web: <http://www.tryton.org/>

IRC: #tryton y #tryton-es en irc.freenode.net

tryton-ar en Google Groups

The logo for Tryton Argentina is a stylized, multi-faceted geometric shape. It features a central white vertical element, with light blue and dark blue triangular facets extending from it. The word 'Tryton' is written in a large, bold, sans-serif font, with 'Argentina' in a smaller, lighter blue font below it. The entire logo is set against a dark blue background.

Tryton Argentina

<https://github.com/tryton-ar>

account_ar: Plan de cuentas de para empresas Argentinas.

account_coop_ar: Plan de cuentas para cooperativas de Trabajo.

party_ar: Integracion del padron AFIP.

account_invoice ar: Facturación integrada con ws de la AFIP.

account_check_ar: Manejo de cheques

account_voucher_ar: Comprobantes de pago.

account_retencion_ar: Retenciones

bank_ar: Incorpora los datos de bancos de Argentina.

cooperative_ar: Gestion de Cooperativa (Recibos, Vacaciones, Socios, etc)

subdiario: Reportes que generan el subdiario

citi_afip: Informativo de Compras y Ventas

PR EG VNT AS!!!

