# odoo

# Technical Training

AGENDA

# Technical Training Overview

## Ten Days of Technical Training

**This course is designed to help partners, integrators, and developers understand the core concept of the Odoo development process. This course is ideal for developers new to the Odoo environment or for IT professionals eager to learn more about the technical aspects of Odoo.**

### Before attending this course, participants must:

· Understand Python - If you do not know Python, visit: **A Byte of Python**

· Have access to Odoo.sh - If you do not have a Odoo.sh subscription code, reach out to your account manager or customer success manager

· Basic Git/GitHub Knowledge - If you are not familiar, visit: **GitHub Guides**

### Having attended this course, participants should be able to:

· Setup and Administer Odoo.sh

· Understand the development concepts and architecture

· Develop a new Odoo module

### The training covers the following technical aspects of Odoo and enables you to:

· Learn how to use Odoo.sh

· Define data models using the ORM framework

· Define menus, list and form views

· Extend existing data models and views

· Define computed fields, default values, constraints

· Define advanced views (calendar, graph, pivot, etc.)

· Define data and actions for models(objects)

· Define custom reports

· Set up security rules (groups, access control list, and record rules)

· Define web controllers

· Support for internationalization (multiple languages)

· Provide overview of web services

· Create Mexican Localization: Addendas
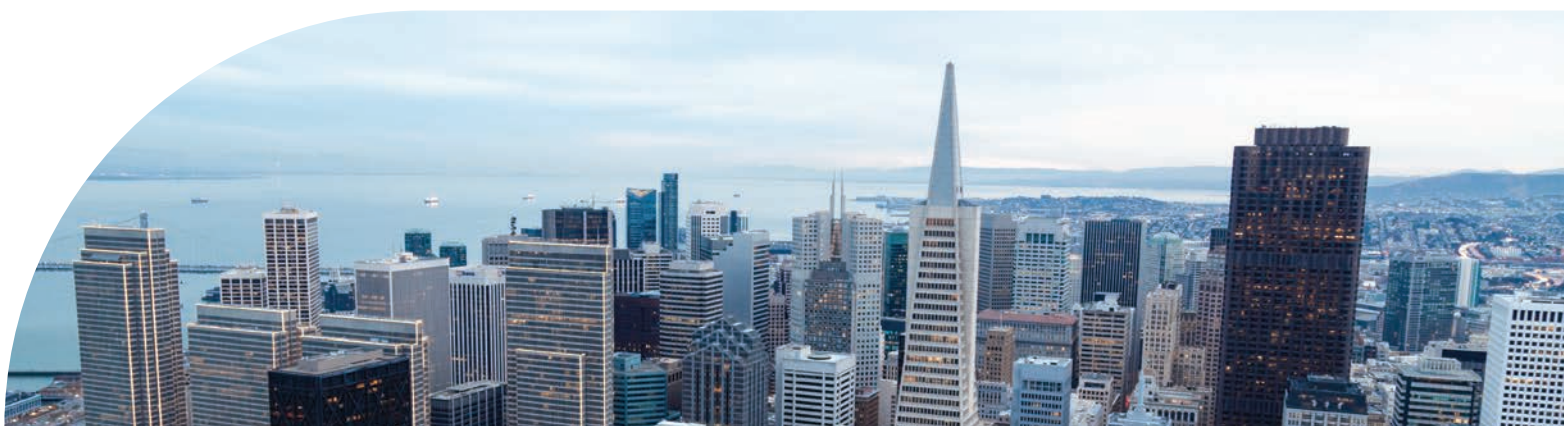
# Daily Agenda

## Day 1

- Odoo.sh Introduction & Setup
- Module Structure
- Architecture

## Day 2

- Models
- Fields
- ORM Basics
- Recordsets
- Datafiles

## Day 3

- Views
- View Structure
- View Fields and Attributes
- List, Form, Search Views

## Day 4

- Actions and Menuitems
- Module Security
- Record Rules
- Groups
- Access Rights

## Day 5

- Advanced ORM
- Method Decorators
- Common Methods
- Exceptions

## Day 6

- Relational Fields
- Many2one, One2Many, Many2many Fields
- Computed Fields
- Related Fields

## Day 7

- Advanced Views (Kanban, Calendar, Map, etc.)

## Day 8

- Model Inheritance
- View Inheritance
- Wizards
- Transient Models

## Day 9

- QWeb Templating Language
- QWeb Reports
- Web Controllers

## Day 10

- XML-RPC External API
- Model Translation
- Report Translation
- **Extra - Mexican Localization Addendas**

# In-depth Agenda

## Setup and Administer Odoo.sh
- Sign in
- Creating Project
  - Github Repository, Odoo Version, Subscription Code, Hosting Location
- Branches
  - Development, Staging, Production
  - Different Branch Tabs
  - Shell Commands
- Builds, Status and Settings Pages

## Odoo Architecture
- Three Tier Architecture
  - Multitenancy
- Database, Application, Presentation
- Model View Controller(MVC)
- PostgreSQL, Odoo Server, Web
- Browser Client

## Odoo Module Structure
- Defining Odoo Modules
  - Manifest and Import files
- Module Composition Directories

## Odoo Models
- Model
  - Business objects as Python classes extending Model
- Model Attributes

## Basic Fields
- Model Fields
- Field Attributes
- General Field Types
  - Boolean, Char, Float, etc.
- Advanced Field Types
  - Binary, Monetary, Date/Datetime, etc.
- Automatic and Reserved Fields

## Object Relational Mapping(ORM) Basics
- ORM for Security and Extensibility
- Recordsets
  - Interactions with models and records
- Environment
- Datafiles
  - Core operations and Structure

## Security
- Access control Mechanisms
  - Access Rights and Record Rules
- Groups
- Field Access

## Actions and Menus
- Triggering Actions
- Window Actions, Server Actions,
- Client Actions, etc.
- Menuitems

## Views
- View Definition and Attributes
- List View, Form View, and Search View

## Advanced ORM
- Method Decorators
- Common ORM Methods
  - Create/Update, Search/Read, Unlink, Recordset Operations
- Odoo Exceptions

## Advanced Fields
- Relational Fields
  - Many2one, One2many, Many2many Fields
- Computed Fields
- Related Fields

# In-depth Agenda (cont.)

## Advanced Views
- Different Advanced View Types
  - Kanban, Calendar, Pivot, Cohort, Gantt, Graph, Activity, Dashboard, Diagram, Map

## Inheritance
- Model Inheritance
  - Classical Inheritance, Extension Inheritance, Delegation Inheritance
- View Inheritance

## Wizards
- Transient Models
- Uses for Wizards and their Properties
- Launching Wizards

## QWeb and Reports
- QWeb Templating Engine
- QWeb Template Directives
  - Conditionals, Loops, Attributes, etc.
- Custom Reports

## Web Controllers
- Controller Class
- Overriding Controllers
- API for Controllers
  - Routing, Request Object, Response Object

## External API
- XML-RPC External API
- Configuration and Connection to Database
- Calling Odoo Methods

## Translation
- Transifex Translation
- Module Translation
  - Explicit/Implicit Export Translation
- Report Translation

## Mexican Localization Addendas
- Addenda Creation
- Addenda Flag Field

# Odoo.sh

## About Odoo.sh

Odoo.sh allows you to easily test your developments, manage your feature-branches and beta-test your features on duplicates of your customer's production instance.

## For Developers

### GitHub Integration

Every commit, pull request, merge or fork is tested, and deployed automatically.

### Continuous Integration

Get a dedicated "runbot" for your project; a dashboard of all tests.

### Clear Logs

Get detailed and filtered logs available in the browser, in real time.

### SSH

Register your public key and connect to any server in just a few clicks.

### Web Shell

Get a shell access to a production server or a container related to a build, in one click.

### Mail Catcher

Mails are deactivated on staging & dev branches. We provide a mail catcher to read them.

### Modules Dependencies

Manage dependencies with third party modules with no pain; update when you want.

## For Testers

### Automated Tests

Every commit goes through a battery of thousands of automated tests.

### Manual Tests

Connect on any feature branches once they are pushed in GitHub, with our instant deployment.

### Staging Branches

Staging branches are built with production data, and stay alive a few weeks for testing.

### Community Modules

Install community modules to test them, in just a few clicks (based on Git submodules).

### Track Developments

Get a detailed history and logs on all development branches to track progress in real time.

# Odoo.sh (cont.)

## For Project Managers

### Dev Staging

Drag development branches to staging, to test them with production data.

### Staging Production

Deploy a staging branch to production in just a drag & drop, once it's fully tested.

### Convenience

Kick off your Odoo project, with a development and hosting platform in three clicks.

### Share Test Builds

Share your builds with your customer to ease testing (public or private URLs).

## For System Administrators

### High Availability

We manage all servers for you: monitoring, backups, emails, dns, ci, staging & production servers.

### Monitoring

Get a status of all your servers, as well as KPIs about their availability and performance.

### Incremental Backups

Incremental backups on 3 different data centers, done twice a day.

### Instant Recovery

Recover any backup in just a few clicks, in a production or staging branch.

### Mail Servers

Production or development mail servers are setup automatically for you.

### DNS

Use your own domain for production server, and our DNS subdomains for development branches.

### Great Performance

PostgreSQL and Odoo are optimized for top maximum performance.

### Top Notch Security

Visit **www.odoo.com/page/security** for further details.

# Odoo's Technical Training

If you have any additional questions, you can reach out to our client solution developer, **Cindey Caine** at **cic@odoo.com**

## CONTACT US

**Americas**
+1 (650) 691-3277

**APAC**
+852 39 500 600

**EMEA**
+32 2 290 34 90

facebook.com/odoo

twitter.com/odoo

linkedin.com/company/odoo

youtube.com/odooapps

instagram.com/odoo.official